Chapter 9

Data Acquisition

Microcontrollers offer a complete signal-chain on a chip for a wide range of applications. One of the most important interfaces between the microcontroller and the real word is the Analogue-to-Digital Converter (ADC). This allows a digital representation of a physical signal to be measured, usually an electrical signal and measured in volts. Typically, the low amplitude of most analogue signals representing physical quantities, such as temperature, humidity, pressure, velocity among others, require some form of signal conditioning. The first stage in this process is often amplification of the analogue signal. This chapter starts by describing the operational amplifiers built into the MSP430 family of devices, their architectures and operation.

To convert an analogue signal to a digital value, it is necessary to use an ADC. The Successive Approximation Register (SAR) converter determines the digital word by approximating the input signal using an iterative process. The Sigma Delta (SD) converter determines the digital word by sampling and digital filtering.

The on-board analogue comparator provided by the MSP430 is configurable, which allows mapping of external pins by inputs or outputs, with interrupt capabilities on either the rising or falling edges.

Finally, at the end of the chapter there are laboratory exercises, which develop applications that group together the use of the operational amplifier and the ADC.

Topic **Page**

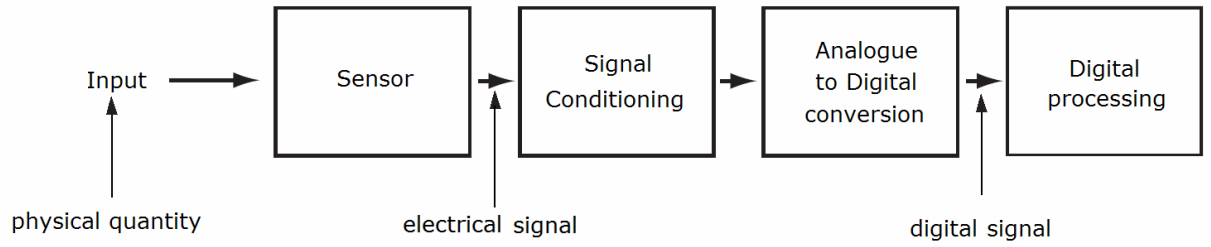
9.1 Data Acquisition Introduction

Nearly all engineering applications require some form of measuring, controlling, calculating, communicating and recording of data. These operations, grouped or isolated, are inherent in measurement instrumentation. If the equipment is to be used for the quantitative analysis of an analogue signal, i.e., a naturally occurring signal, the following must be taken into consideration:

- Measuring method;
- Data recording method;
- Interaction between the different equipment and between the analogue signal source;
- Admissible error margin;
- Noise and interference influence;
- Measuring uncertainty;
- Type of control resulting from the measurement (analogue/digital);
- Method of communication with other equipment;
- Mathematical capacity;
- …

The analogue signal to be measured may be temperature, pressure, humidity, velocity, flow rate, linear motion, position, amongst others. This signal must be converted into an analogue electrical signal, typically voltage or current, and then into a digital form that can be processed by an electronic circuit. The first task requires sensors to convert the physical quantities into electrical signals. Generally, sensors convert physical quantities into analogue electrical signal in the range of millivolts or milliamps.

Figure 9-1. Data acquisition block diagram.



Signal conditioning relates to the operations required to convert the analogue electrical signal measured by the sensor to the signal level supported by analogue-to-digital converter (ADC). This typically involves operations such as signal filtering and amplification. Signal filtering may reduce the signal level, but the amplification operation is linear, so that the output maintains all its characteristics, being changed only in amplitude.

When the analogue electrical signal is conditioned to be compatible with the range of values supported by the ADC, the conversion operation initiates a sample-and-hold function. This takes a snapshot of the continuously changing input signal and holds on to it until the next sample is acquired. Note that the Sample-and-hold is not necessary for Sigma-Delta (SD) converters, nor for slope converters, nor for all flash converters and is automatically implemented as part of structure of capacitive Successive Approximation Register (SAR) converters on the MSP430. The specifications of these converters will be described in the following sections.

The time interval between samples should be based on the Nyquist theorem, so that the analogue signal is converted into a digital signal that reproduces all its amplitude variations. This procedure requires a balance between the speed of the conversion process and the sampling rate, in order to minimize the error between the true input voltage and the ADC output voltage measured. The resolution of the ADC needs to be sufficient to give the required digital signal accuracy. The output value of the sample-and-hold is fed into the ADC, which generates a digital code that can be used by a digital processing system.

Several MSP430 devices include on-chip the signal conditioning and analogue-to-digital converters. Additionally, some of the devices include an internal temperature sensor.

The following section describes the operational amplifier topologies and different types of ADC, as contained in MSP430 devices, as well as their configurations and applications.

9.2 Operational Amplifiers

Some devices in the MSP430 family provide signal amplification features to avoid the need for an external analogue operational amplifier (Op-Amp) circuit for signal conditioning. The main Op-Amp characteristics are:

- Signal protection from interference (voltage level transients);
- Good signal transfer due to high impedance inputs and low impedance outputs;
- Improvement of signal precision, due to conversion of the input signal to the voltage level required by the ADC.

There are several different types of Op-Amp architectures

- Single Supply;
- Rail-to-rail In;
- Rail-to-rail Out;
- CMOS, Bipolar or mixed;
- Dual Supply.

All Op-Amps (OAs) included in the MSP430 devices are Single Supply and CMOS.

The MSP430FG4618 implemented in the Experimenter's board contains three Op-Amps. The MSP430F2274 included in the eZ-RF2500 contains two Op-Amps.

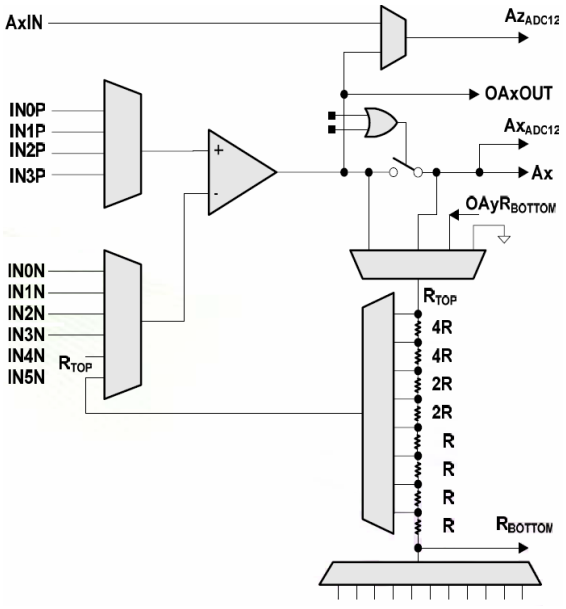
The main Op-Amp features are:

- ❑ Selectable gain bandwidth 500 kHz (current consumption: ~50 μA), 1.4 MHz, or 2.2 MHz;
- ❑ Class AB output for mA range drive;
- ❑ Integrated charge pump for rail-to-rail input range and superior offset behaviour (FG only);
- ❑ User-configurable feedback and interconnections:
  - Internal R ladder;
  - Internally chainable;
    - Avoiding external passive components;
  - Internal connections to the ADC and DAC.

The internal structure of each Op-Amp allows:

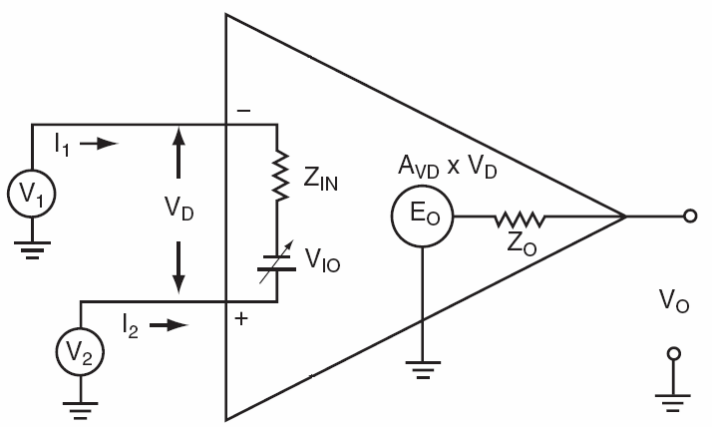
- ❑ Flexible feedback network;
- ❑ Flexible modes, to optimize current consumption and performance;
- ❑ User configurable for:
  - General purpose;
  - Unity gain buffer;
  - Voltage comparator;
  - Inverting programmable gain amplifier (PGA);
  - Non-inverting programmable gain amplifier (PGA);
  - Differential amplifier.

Figure 9-2. Op-Amp internal structure.



The general purpose Op-Amp shown in *Figure 9-3* is characterized by its high gain, high input impedance, low output impedance, and wide bandwidth.

Figure 9-3. General purpose Op-Amp.



An Op-Amp is characterized by:

- Two inputs:
  - Inverting input, V_1;
  - Non inverting input, V_2;
- One output, V_0:
  - Represented by a generator, $E_0 = A_{VD} \times V_D$:
    - E0: input differential signal, $V_D = V_2 - V_1$;
    - A_{VD}: Open-loop differential gain (ideally: infinity, but in practice an op-amp has A_{VD} >20000).
- High input impedance, Z_{IN} (ideally: infinity);
- Low output impedance, Z_0 (ideally: zero);
- Input offset voltage, V_{IO}: the output voltage is displaced from 0 V when there is no differential input signal (ideally: zero);
- Null input currents, I_1 and I_2 (ideally: zero).

The amplification topology and characteristics of an Op-Amp vary depending on the passive external components and connections (MSP430 also include some of these components internally).

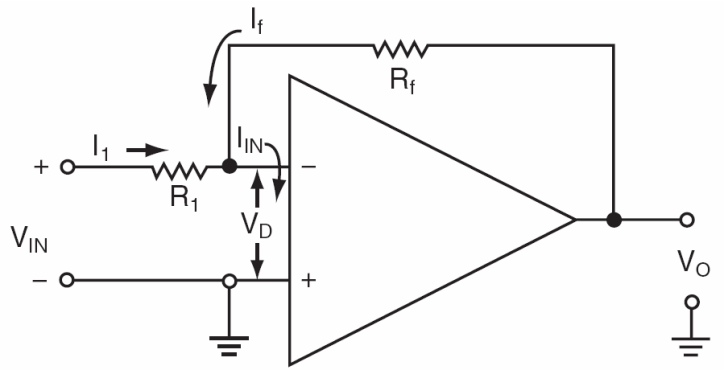
- Signal referred to Op-amp ground:
  - Inverting topology;
  - Non-inverting topology;
  - Buffer.
- Signal referred to a different potential to Op-Amp ground:
  - Differential amplifier: Proportional to the difference in voltage between the two inputs: $V_0 = A_{VD}(V_2 - V_1)$.

9.2.1 Architectures of operational amplifiers

Inverting topology

- Resistor R_f is connected from the output back to the inverting input, to control the gain of the Op-amp using negative feedback;
- V_{IN} applied to the inverting input;
- Gain of the inverting Op-Amp: $A_{VD} = - R_f / R_1$.

Figure 9-4. Inverting Op-Amp topology.



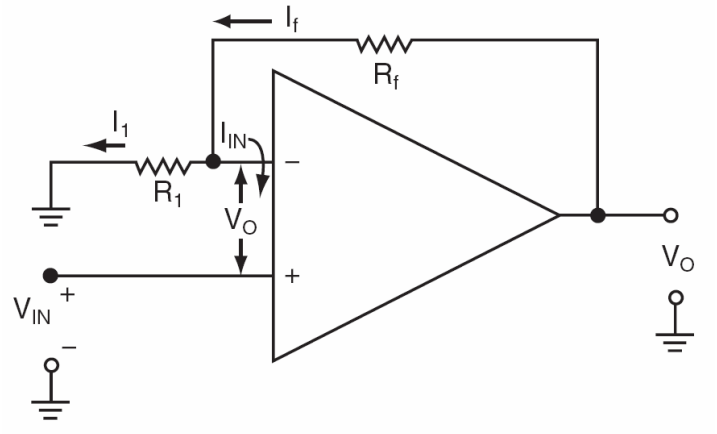
Note: the single supply circuitry shown is only applicable for negative input voltages, and the input signal is loaded by R_1.

- Characteristics:
  - The output has a 180º phase shift with respect to the input;

Non-inverting topology

- Resistor R_f is connected from the output back to the inverting input, to control the gain of the op-amp with negative feedback;
- V_{IN} applied to the non-inverting input;
- Gain of the non-inverting Op-Amp: $A_{VD} = 1 + R_f / R_1$.

Figure 9-5. Non inverting Op-Amp topology.

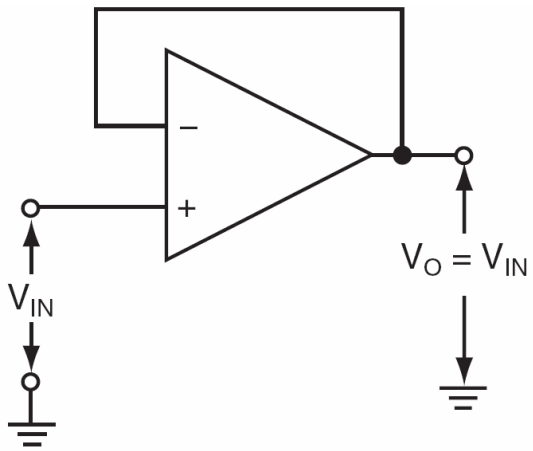


- Characteristics:
  - Output in phase with the input;
  - Buffer (isolation between the circuit and the charge);
  - Power amplifier;
  - Impedance transformer;
  - Input impedance: 5×10^5 to 1×10^{12} Ω;
  - Useful topology to amplify a signal from a source, which has a high source impedance, so that minimum current is taken.

Unity gain buffer (voltage follower) topology

- Non-inverting amplifier with $R_f = 0$ (Note: often used with R_f for better dynamic performance) and R_1 equal to infinity;
- $A_{VD} = 1 + R_f/R_1 = 1$ (unity gain amplifier);
- $V_0 = V_{IN}$.

Figure 9-6. Buffer Op-Amp topology.



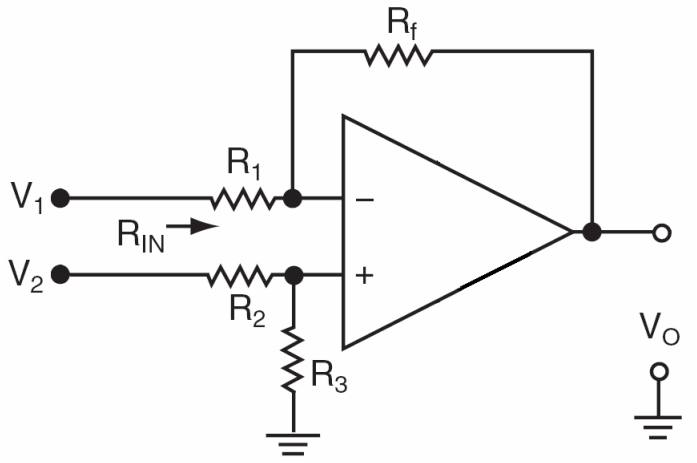
Differential topology

- Combination of inverting and non-inverting topologies;
- Output signal is an amplified version of the difference between the two input signals;
- $A_{VD} = R_f/R_1$;
- $V_0 = A_{VD}(V_2 - V_1)$;
- Common-Mode Rejection Ratio (CMRR):
  - Noise voltage picked up on the leads connecting the sensor to the amplifier may be 100 to 1000 times greater than the output signal of the sensor;
  - CMRR ensures that any signal appearing on both input lines at the same time will not appear at the output;
  - CMRR [dB] = $20\log_{10}(A_{VD}/A_{CM})$;

    where: A_{CM}: Amplification for Common Mode;

    $A_{CM} = (R_1 R_3 - R_f R_2) / [R_1 (R_2 + R_3)]$.

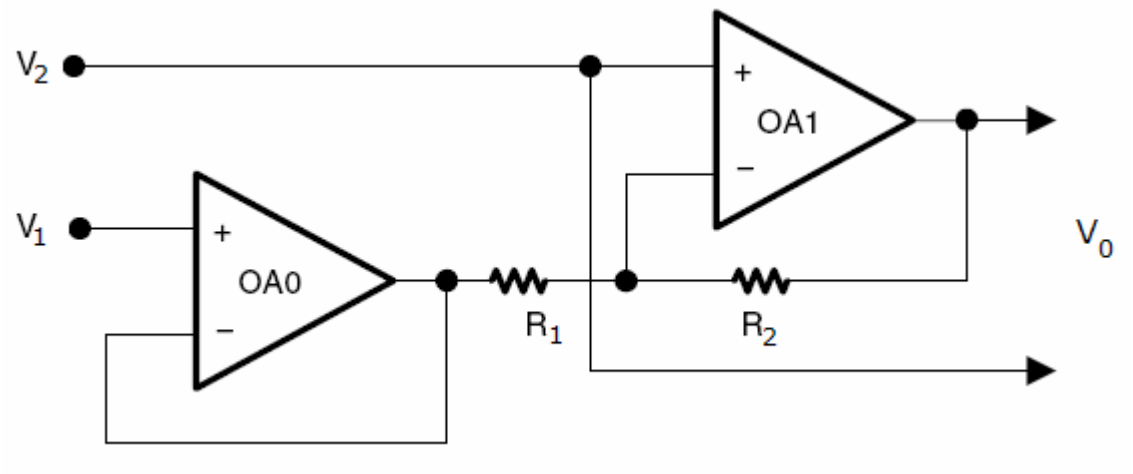
Figure 9-7. Differential Op-Amp topology.



Two Op-Amp Differential topology

- Combination of inverting and non-inverting topologies;
- Output signal is the amplification of the difference between the input signals;
- $A_{VD} = R_2/R_1$;
- $V_0 = A_{VD}(V_2 - V_1)$;

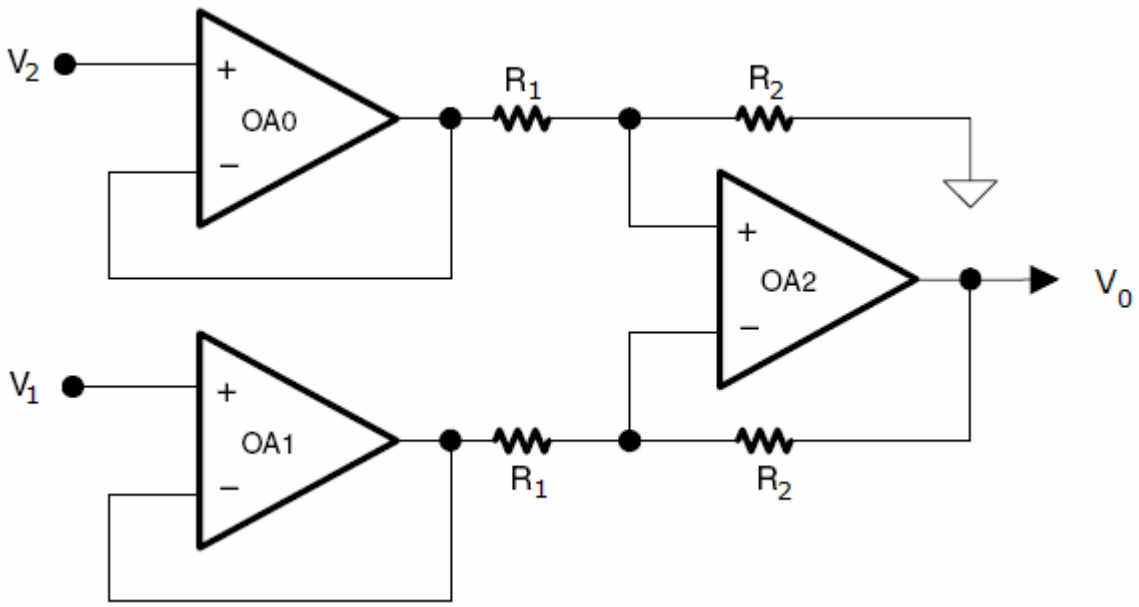
Figure 9-8. Two Op-Amp Differential topology.



Three Op-Amp Differential topology

- Combination of inverting and non-inverting topologies;
- The output signal is an amplified version of the difference between the two input signals;
- $A_{VD} = R_2/R_1$;
- $V_0 = A_{VD}(V_2 - V_1)$;

Figure 9-9. Three Op-Amp Differential topology.



9.2.2 Operational amplifiers registers

OAxCTL0, Op-Amp Control Register 0

7	6	5	4	3	2	1	0
OANx		OAPx		OAPMx		OAADC1	OAADC0

Bit		Description
7-6	OANx	OA Inverting input signal select: OAN1 OAN0 = 00 ⇒ OAxI0 OAN1 OAN0 = 01 ⇒ OAxI1 OAN1 OAN0 = 10 ⇒ DAC0 internal OAN1 OAN0 = 11 ⇒ DAC1 internal
5-4	OAPx	OA Non-inverting input signal select: OAP1 OAP0 = 00 ⇒ OAxI0 OAP1 OAP0 = 01 ⇒ OAxI1 OAP1 OAP0 = 10 ⇒ DAC0 internal OAP1 OAP0 = 11 ⇒ DAC1 internal
3-2	OAPMx	Selection of the slew rate vs. current consumption for the OA: OAPM1 OAPM0 = 00 ⇒ Off OAPM1 OAPM0 = 01 ⇒ Slow OAPM1 OAPM0 = 10 ⇒ Medium OAPM1 OAPM0 = 11 ⇒ Fast
1	OAADC1	OA output select (OAFCx > 0): OAADC1 = 1 ⇒ OAx output connected to internal /external A1 (OA0), A3 (OA1), or A5 (OA2) signals
0	OAADC0	OA output select (OAPMx > 0): OAADC0 = 1 ⇒ OAx output connected to internal A12 (OA0), A13 (OA1), or A14 (OA2) signals

OAxCTL1, Op-Amp Control Register 1

7	6	5	4	3	2	1	0
OAFBRx			OAFCx			Reserved	OARRIP

Bit		Description
7-5	OAFBRx	OAx feedback resistor: OAFBR2 OAFBR1 OAFBR0 = 000 ⇒ (Gain): A_{VD} = 1 OAFBR2 OAFBR1 OAFBR0 = 001 ⇒ (Gain): A_{VD} = 1.33 OAFBR2 OAFBR1 OAFBR0 = 010 ⇒ (Gain): A_{VD} = 2 OAFBR2 OAFBR1 OAFBR0 = 011 ⇒ (Gain): A_{VD} = 2.67 OAFBR2 OAFBR1 OAFBR0 = 100 ⇒ (Gain): A_{VD} = 4 OAFBR2 OAFBR1 OAFBR0 = 101 ⇒ (Gain): A_{VD} = 4.33 OAFBR2 OAFBR1 OAFBR0 = 110 ⇒ (Gain): A_{VD} = 8 OAFBR2 OAFBR1 OAFBR0 = 111 ⇒ (Gain): A_{VD} = 16
4-2	OAFCx	OAx function control: OAFC2 OAFC1 OAFC0 = 000 ⇒ General purpose OAFC2 OAFC1 OAFC0 = 001 ⇒ Unity gain buffer OAFC2 OAFC1 OAFC0 = 010 ⇒ Reserved OAFC2 OAFC1 OAFC0 = 011 ⇒ Comparing Op-Amp OAFC2 OAFC1 OAFC0 = 100 ⇒ Non-inverting PGA OAFC2 OAFC1 OAFC0 = 101 ⇒ Reserved OAFC2 OAFC1 OAFC0 = 110 ⇒ Inverting PGA OAFC2 OAFC1 OAFC0 = 111 ⇒ Differential Op-Amp
0	OARRIP	OA rail-to-rail input off: OARRIP = 0 ⇒ OAx input signal range is rail-to-rail OARRIP = 1 ⇒ OAx input signal range is limited

9.2.3 Topologies configuration

The OA module can be configured for different amplifier topologies using the OAFCx bits. The different topologies available are given in *Table 9-1*.

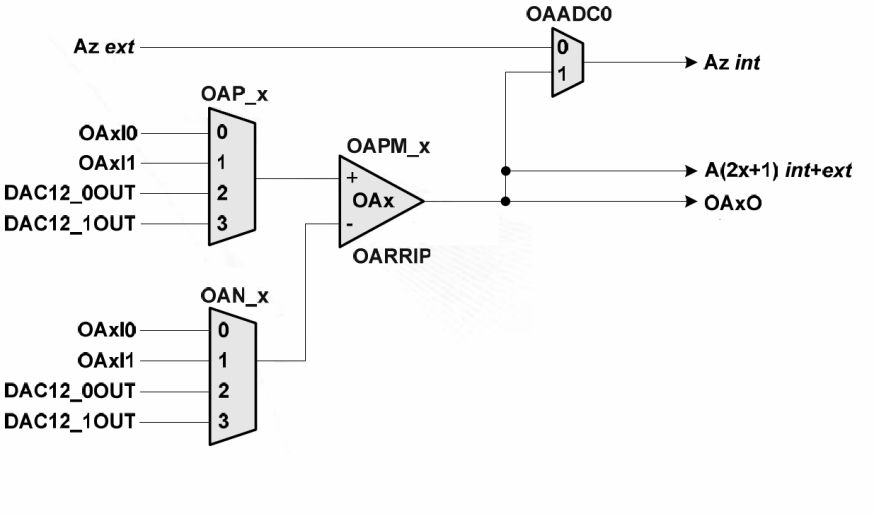
Table 9-1. Op-Amp (OA) module topologies configuration.

OAFCx bits	Op-Amp (OA) module topology
000	General-purpose op-amp
001	Unity gain buffer
010	Reserved
011	Voltage comparator
100	Non-inverting programmable amplifier
101	Reserved
110	Inverting programmable amplifier
111	Differential amplifier

General-purpose Op-Amp (OAFCx = 000)

- Closed loop configuration;
- Connection from output to inverting input;
- The feedback resistor ladder is isolated from OAx:
  - Requires external passive components (resistors) to configure the above-mentioned topologies and required gain.
- OAxCTL0 bits define the signal routing;
- OAx inputs are selected with the OAPx and OANx bits;
- OAx output is internally connected to the ADC12 input channel (selected by the OAxCTL0 bits).

Figure 9-10. General purpose Op-Amp.

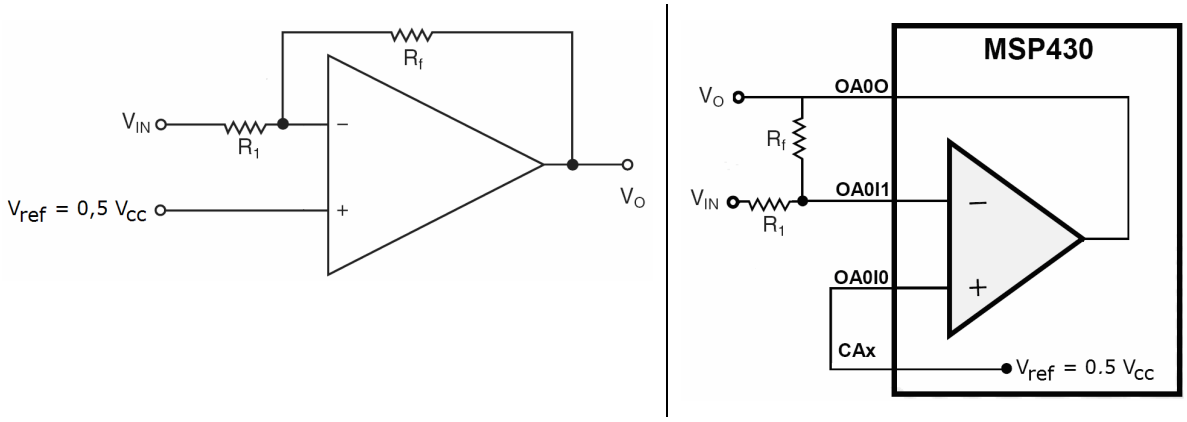


The Op-Amp output OAxO, can generate a port pin interrupt or a timer event.

This mode can provide both inverting and non-inverting amplifier topologies:

- ***Inverting amplifier topology (OAFCx = 110):***

Figure 9-11. General purpose Op-Amp connections – Inverting amplifier topology.

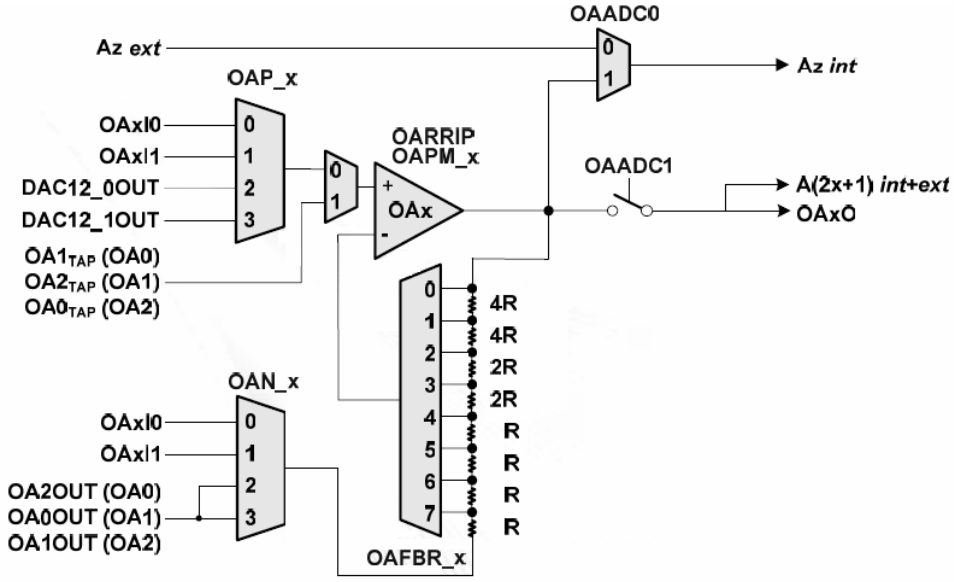


The output voltage is given by the following equation:

$$V_0 = V_{ref}\left(1 + \frac{R_f}{R_1}\right) - V_{IN}\frac{R_f}{R_1}$$

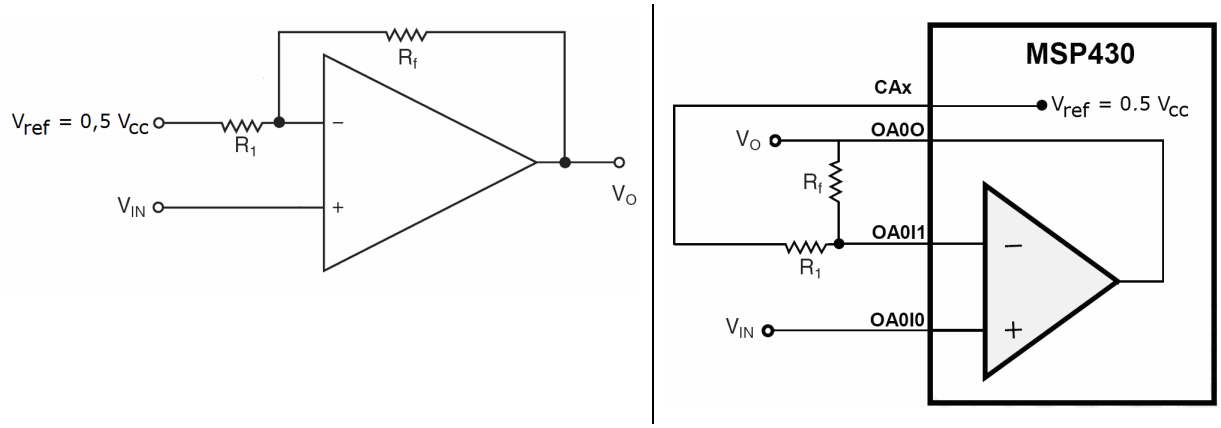
The configuration of the OAxCTL1 register allows the use of internal resistors, giving gain in the range from A_{VD} = -0.33 to A_{VD} = -15. Additionally, the OAx input signal range can be selected to be rail-to-rail or limited by the OARRIP bit setting.

Figure 9-12. Inverting PGA topology.



- ***Non-inverting amplifier topology (OAFCx = 100):***

Figure 9-13. General purpose Op-Amp connections – Non-inverting amplifier topology.



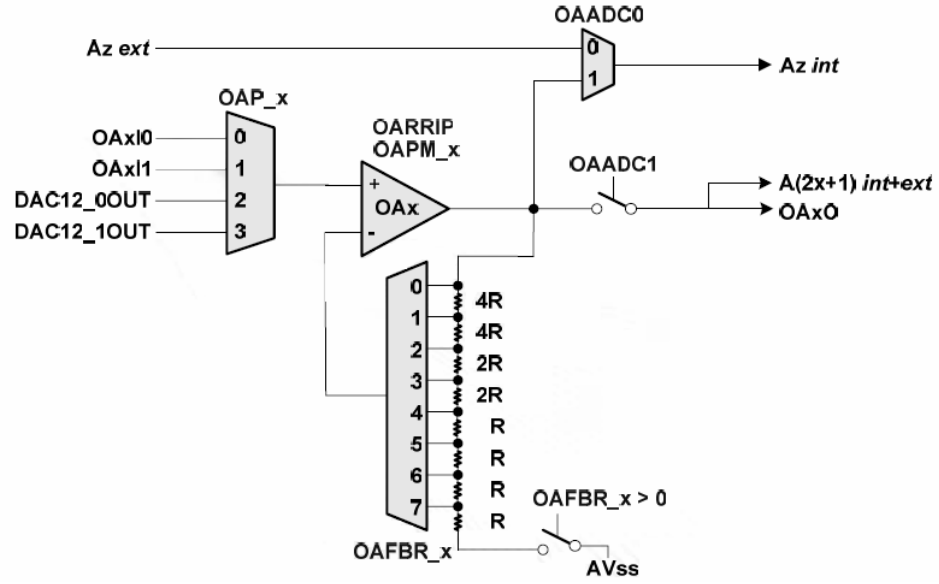
The output voltage is given by the following equation:

$$V_0 = V_{IN}\left(1 + \frac{R_f}{R_1}\right) - V_{ref}\frac{R_f}{R_1}$$

The configuration of the OAxCTL1 register allows to use the internal resistors to give gain in the range from $A_{VD} = 1$ to $A_{VD} = 16$. Additionally, the OAx input signal range can be selected to be rail-to-rail or limited by the OARRIP bit selection.

- OARRIP = 1: (V_{SS} - 0.1V) {min} to (V_{CC} + 0.1) {max}
  - Charge pump at input stage is turned on
- OARRIP = 0: (V_{SS} - 0.1V) {min} to (V_{CC} - 1.2) {max}
  - Appropriate for Gains > 2.

Figure 9-14. PGA Non–inverting amplifier topology.



Unity gain buffer (OAFCx = 001)

- Closed loop configuration;
- OAx output is connected internally to R_{BOTTOM} and to the inverting input of the OAx, providing a unity-gain buffer;
- Non-inverting input is available on a controller pin (selected by the OAPx bits);
- External connection to the inverting input is disabled;
- OAx output is internally connected to the ADC12 input channel (selected by the OAxCTL0 bits).

Figure 9-15. Unity gain buffer.

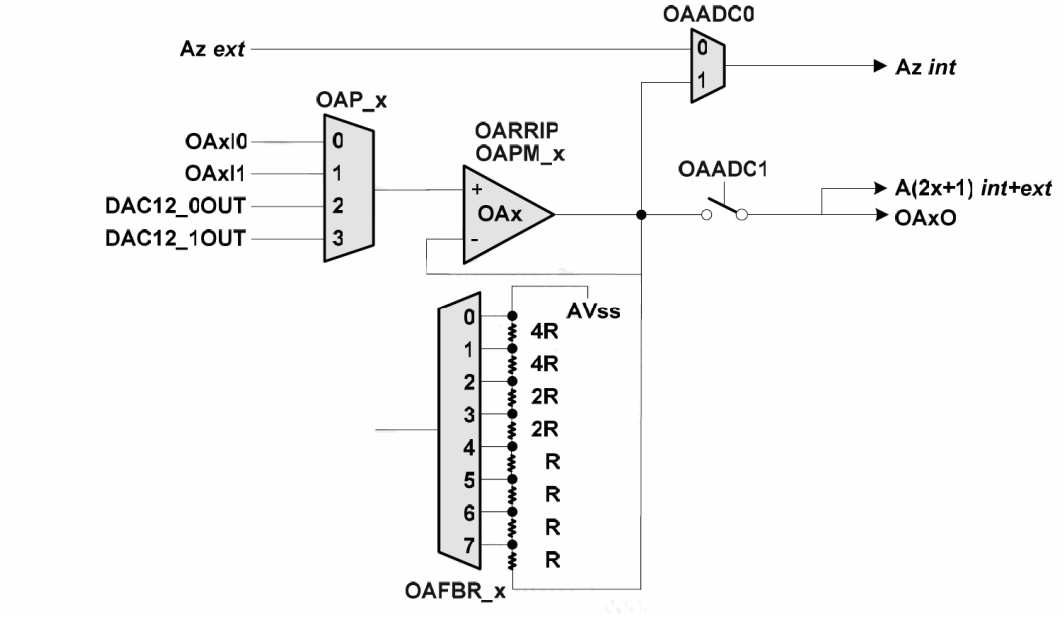
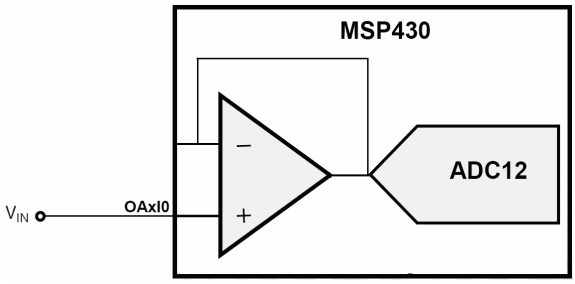


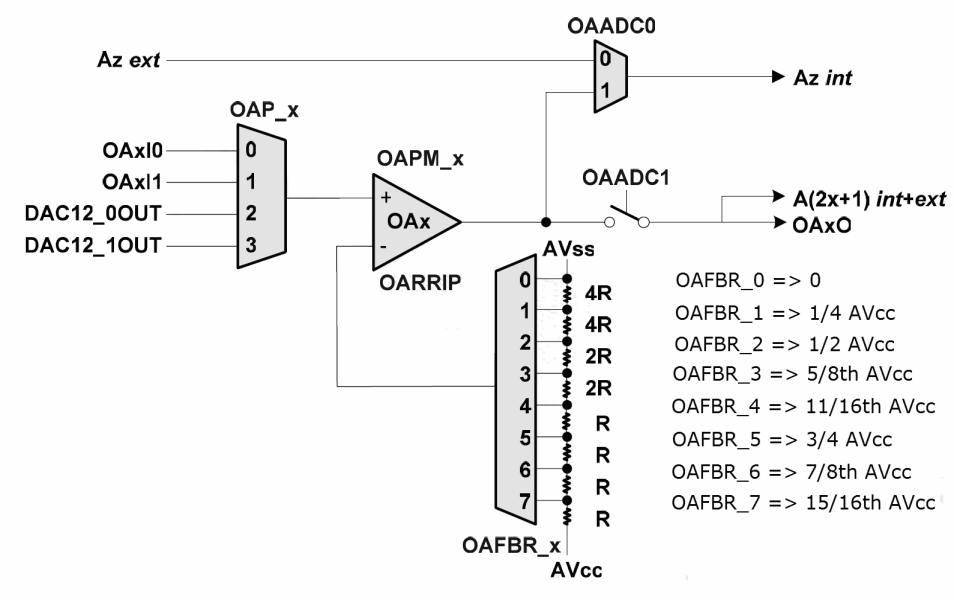
Figure 9-16. Unity gain buffer Op-Amp connections.



Voltage comparator (OAFCx = 011)

- Open loop configuration;
- OAx output is isolated from the resistor ladder;
- R_{TOP} is connected to AV_{SS};
- R_{BOTTOM} is connected to AV_{CC};
- OAxTAP signal is connected to the inverting input of the OAx, providing a comparator with a programmable threshold voltage (selected by the OAFBRx bits);
- Non-inverting input is selected by the OAPx bits;
- Hysteresis can be added by an external positive feedback resistor;
- The external connection for the inverting input is disabled;
- OAx output is internally connected to the ADC12 input channel (selected by the OAxCTL0 bits).

Figure 9-17. Voltage comparator.



Differential amplifier (OAFCx = 111)

This mode allows internal routing of the OA signals for a two Op-Amp or three Op-Amp instrumentation amplifier.

Two Op-Amp differential amplifier

- In a two Op-Amp configuration with OA0 and OA1 (see Figure *9-18*), the output of OAx is connected to R_{TOP} by routing through another OAx in Inverting PGA mode.
- R_{BOTTOM} is unconnected to provide a unity gain buffer. This buffer is combined with the remaining OAx to form the differential amplifier.
- The OAx output is internally connected to the ADC12 input channel, as selected by the OAxCTL0 bits.

The two Op-Amp differential amplifier topology combines OA0 and OA1. It requires the following registers to be configured:

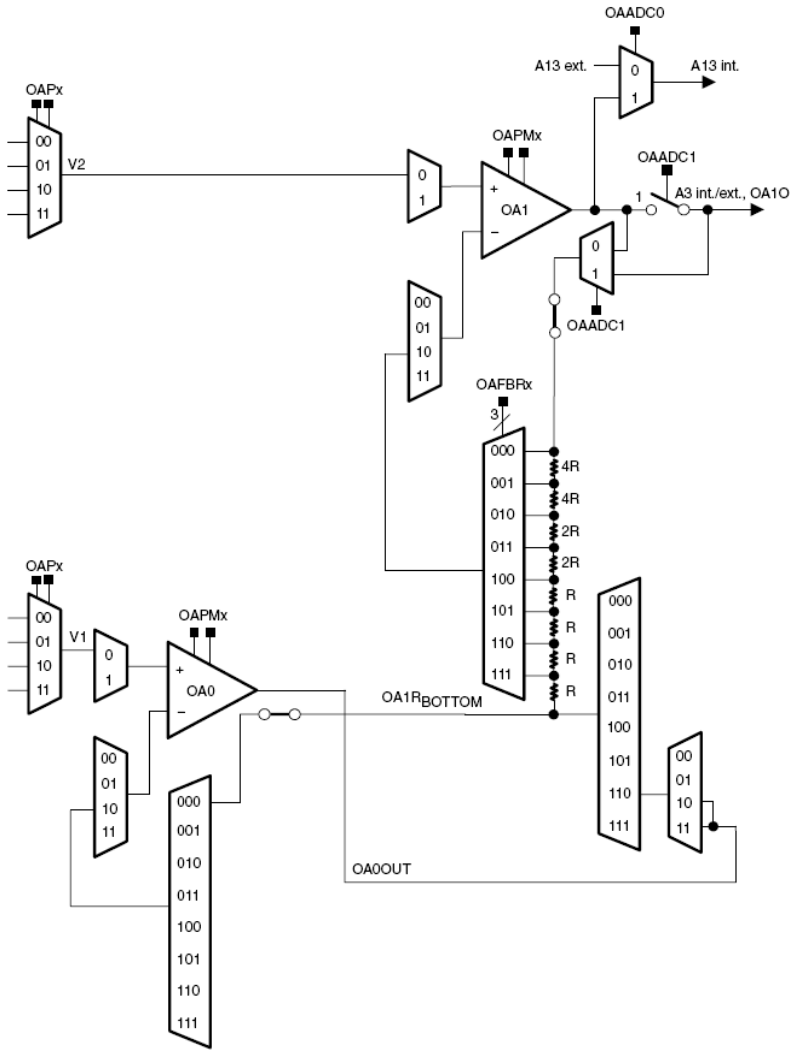
Table 9-2. Two Op-Amp Differential topology control registers configuration.

Registers	Configuration
OA0CTL0	00 xx xx 00
OA0CTL1	00 01 11 0x
OA1CTL0	10 xx xx xx
OA1CTL1	xx x1 10 0x

Table 9-3. Two Op-Amp Differential topology gain configuration.

OA1 OAFBRx bits	Gain
000	0
001	0.33
010	2
011	2.67
100	3
101	4.33
110	7
111	15

Figure 9-18. Two Op-Amp Differential topology.



Three Op-Amp differential amplifier

The three Op-Amp differential amplifier topology (see *Figure 9-19*) combines OA0, OA1 and OA2. It requires the following registers to be configured:

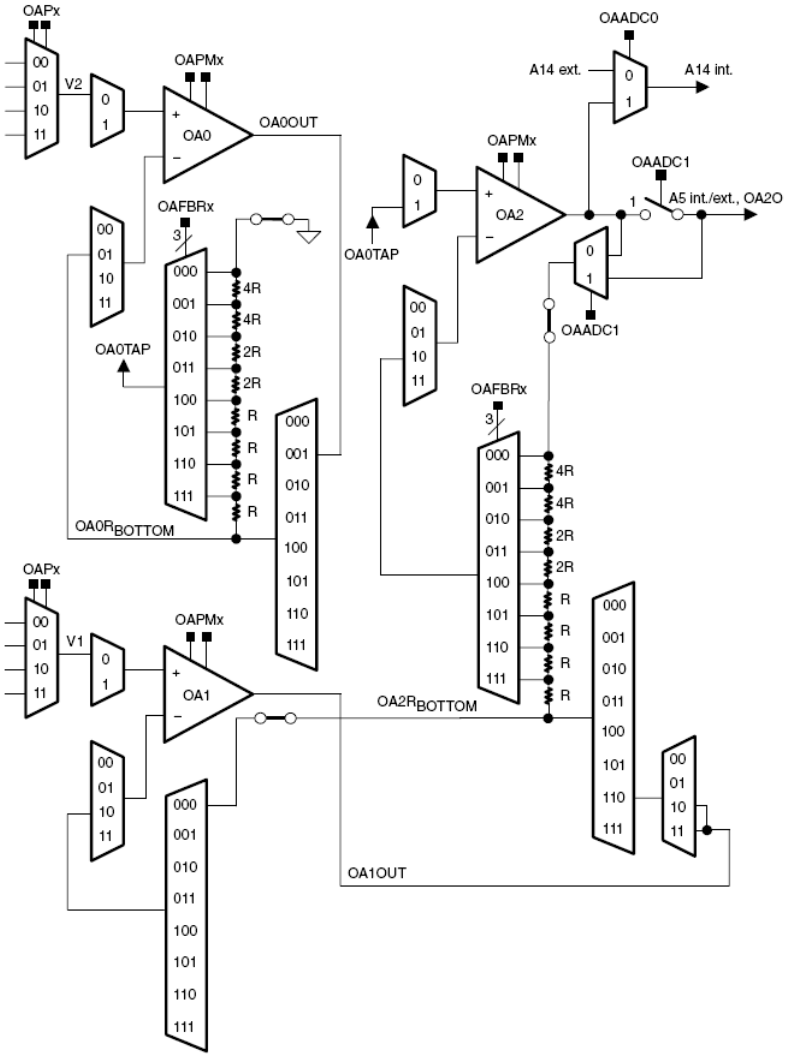
Table 9-4. Three Op-Amp Differential topology control registers configuration.

Registers	Configuration
OA0CTL0	00 xx xx 00
OA0CTL1	xx x0 01 0x
OA1CTL0	00 xx xx 00
OA1CTL1	00 01 11 0x
OA2CTL0	11 11 xx xx
OA2CTL1	xx x1 10 0x

Table 9-5. Three Op-Amp Differential topology gain configuration.

OA0/OA2 OAFBRx bits	Gain
000	0
001	0.33
010	2
011	2.67
100	3
101	4.33
110	7
111	15

Figure 9-19. Three Op-Amp Differential topology.



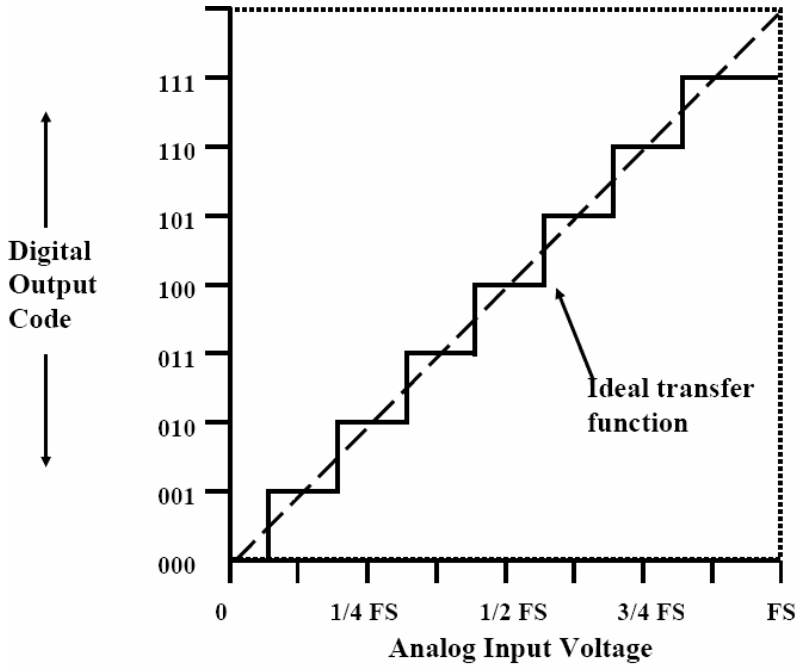
9.3 Analogue-to-Digital Converter (ADC)

The following sections give the ADC architectures of the MSP430 devices, and how to use them with the hardware development tools.

The analogue world (the real one), interfaces with the digital systems through an ADC. This takes the voltage from the transducer (after signal conditioning) as an input, and converts it to an equivalent digital value. The ideal ADC transfer function (3 bit ADC) is shown in *Figure 9-20*. This digital value can then be displayed, processed, stored or transmitted.

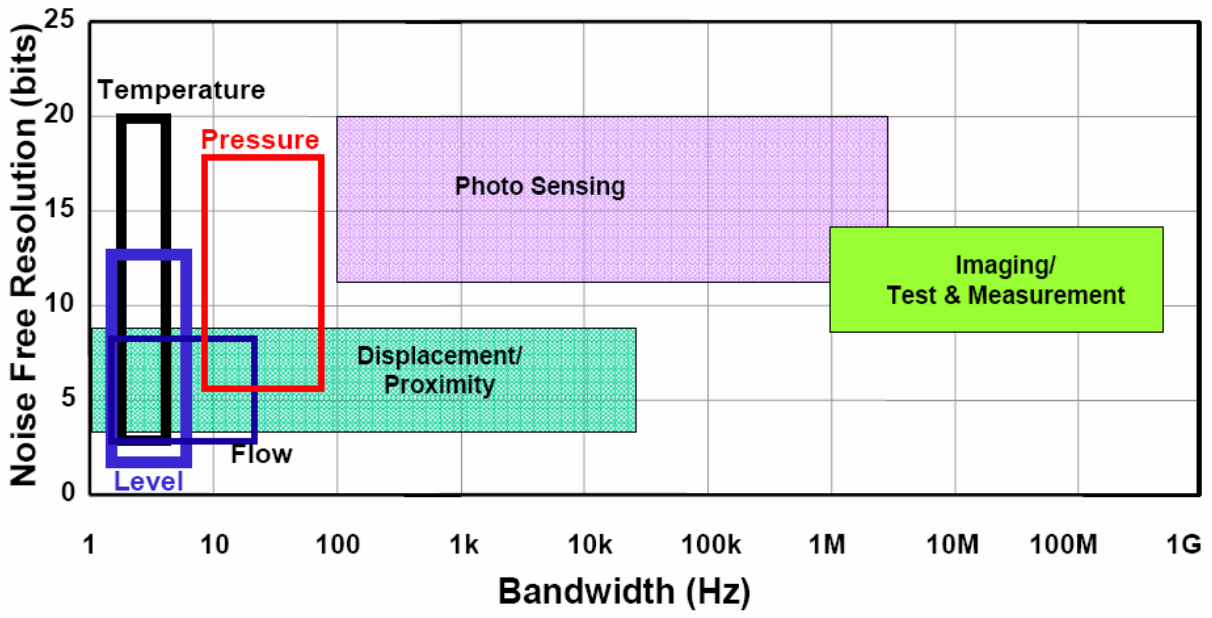
All analogue voltages between zero and full scale of the ADC must be quantized, by dividing the range of voltage into sub-ranges. If *FS* is the full-scale analogue voltage, the quantization increment is given by *FS* x *LSB*, where $LSB = 2^{-n}$, where *n* is the number of bits of the ADC. The quantization process, which replaces a linear analogue function with a staircase digital representation, results in a quantization uncertainty of ± 0.5 LSB and a quantization error.

Figure 9-20. ADC ideal transfer function for a 3 bit ADC.



The analogue peripherals in a number of MSP430 family devices are sufficient to realize complete practical signal chains, with just a few passive components. Furthermore, the processing capabilities of the MSP430 are sufficient to implement some interesting real world signal processing tasks. Each analogue class of applications is more or less defined over a bandwidth range and typically requires an established number of bits for noise-free resolution, as shown in *Figure 9-21*.

Figure 9-21. Analogue classes of applications: Noise free resolution vs. Bandwidth.



9.3.1 ADC specifications

Before presenting the ADC architectures, it is important to define several specifications normally included on ADC datasheets, which are important to the development of the analogue chain.

First it is necessary to discuss the differences between resolution and accuracy, and their influence on ADC performance.

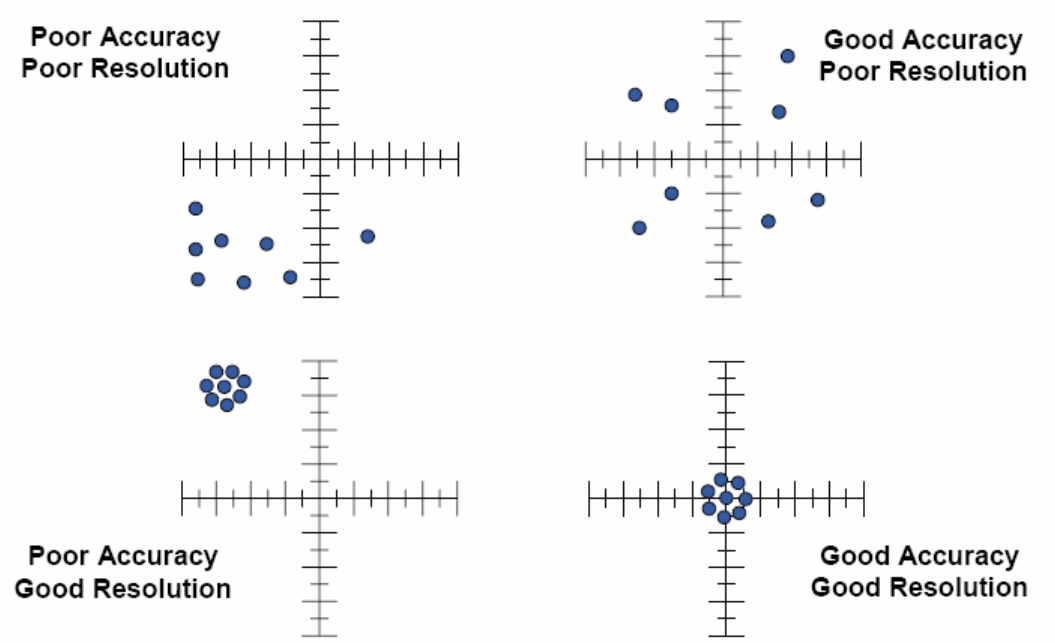
The **resolution**, *R*, of an ADC is the smallest analogue voltage that can be converted into a digital code, that is, the Least Significant Bit (LSB). It can be written as:

$$R = \frac{1}{2^n}$$

The **accuracy** is the degree of conformity of a digital code to it actual (true) analogue voltage. Accuracy can be expressed as the "degree of truth".

Figure 9-22 shows the differences between good/poor resolution and accuracy.

Figure 9-22. Examples of resolution and accuracy levels.



The resolution only specifies the bit size of the digital output value, not the performance.

The performance is expressed by the following specifications:

- Speed;
- Accuracy, which depends on the circuitry type:
  - DC: Integral Non-Linearity (INL), Differential Non-Linearity (DNL), Offset, Gain;
  - AC: Noise, Distortion...

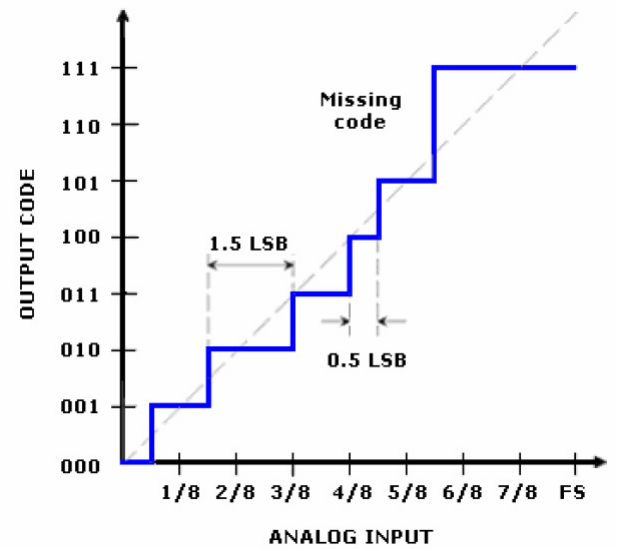
DC performance

- ***Differential Non-Linearity (DNL)***

DNL reveals how far an output code is from a neighbouring output code. The distance is measured as a change in input voltage magnitude and then converted to LSBs.

To be free of DNL error, it requires that as the input voltage is swept over its range, all output code combinations will appear at the converter output. A DNL error of < ± 1 LSB guarantees no missing codes. *Figure 9-23* illustrates an example of the DNL error for a 3-bit ADC.

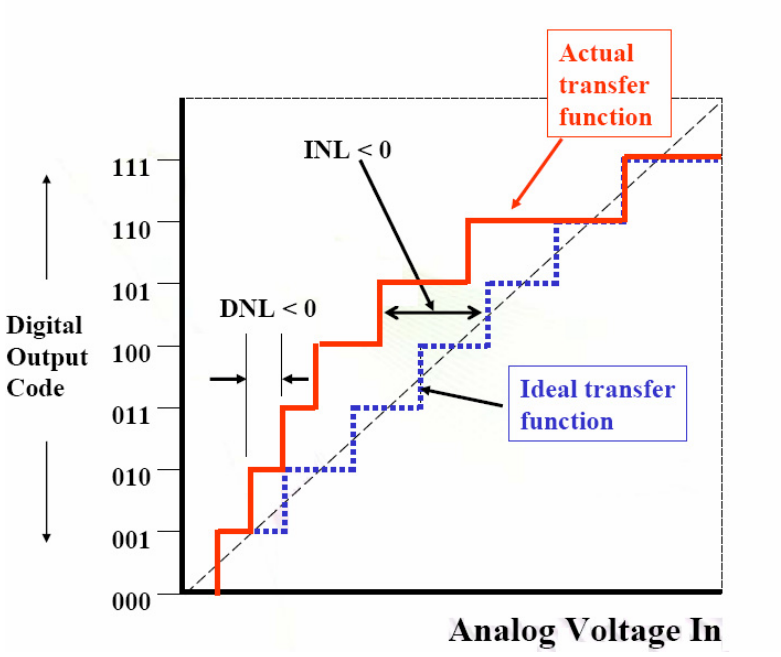
Figure 9-23. Example of the DNL error for a 3-bit ADC.



- ***Integral Non-Linearity (INL)***

INL is defined as the integral of the DNL errors. The INL error represents the difference between the measured converter result and the ideal transfer-function value. *Figure 9-24* shows both DNL and INL errors for a 3-bit ADC.

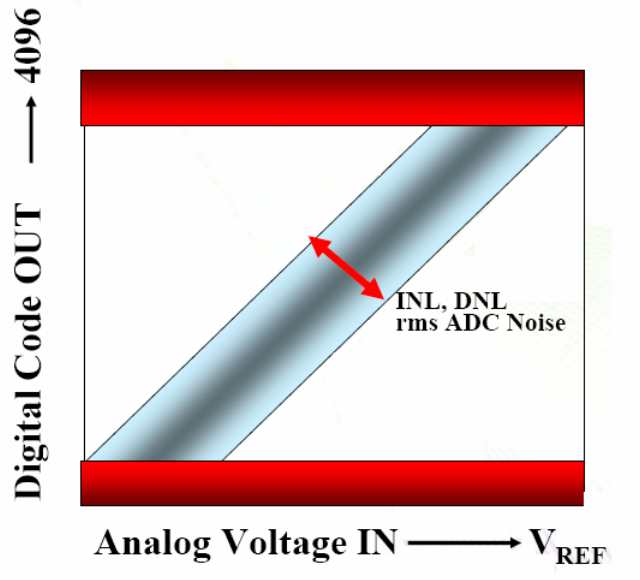
Figure 9-24. Example of DNL and INL errors for a 3-bit ADC.



DNL, INL and noise will impact on the dynamic range:

- ❑ INL, DNL and Noise errors act across the entire range, as shown in *Figure 9-25*;
- ❑ Impacts the Effective Number of Bits (ENOB);
- ❑ Not easily calibrated or corrected;
- ❑ Effects accuracy.

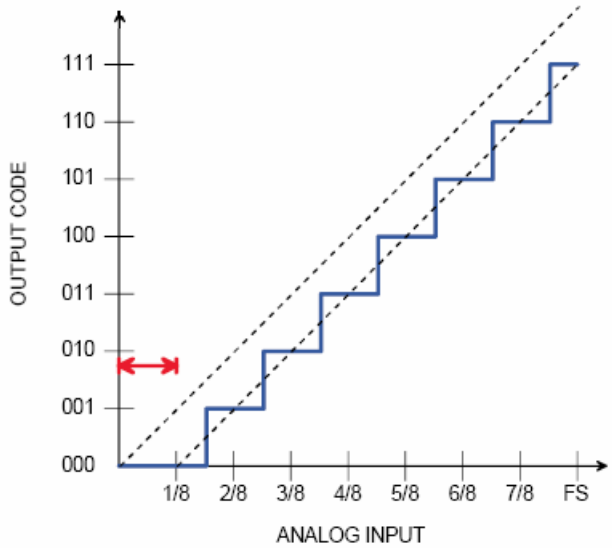
Figure 9-25. DNL, INL and RMS noise impact on the dynamic range of a 12-bit ADC.



- ❑ ***Offset error***

In bipolar systems, offset error (see *Figure 9-26*) shifts the transfer function, but does not reduce the number of available codes.

Figure 9-26. Offset error on a 3-bit ADC.



- ***Gain error***

The gain error is given by the full-scale error, minus the offset error. A comparison of the gain error and an ideal ADC transfer function is shown in *Figure 9-27*.

Note that the gain error specification may, or may not include errors contributed by the ADC reference voltage (see *Figure 9-28*).

Figure 9-27. Gain error on a 3-bit ADC.

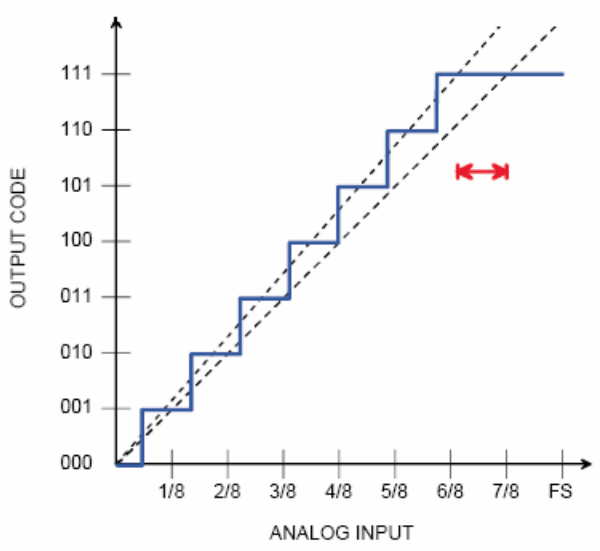
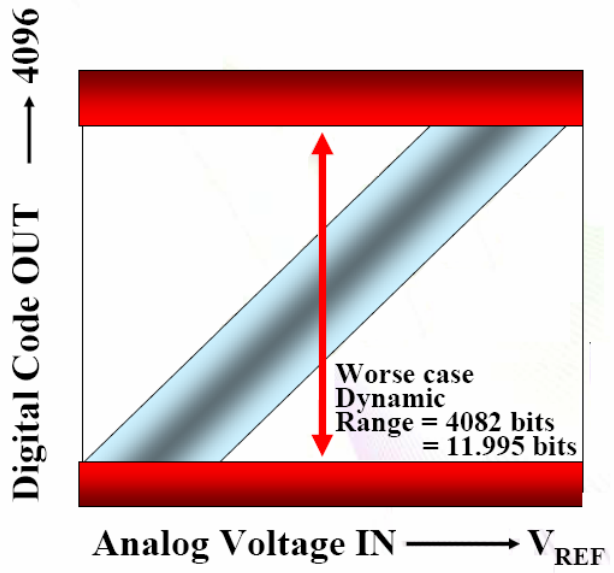


Figure 9-28. Offset and gain errors impact on the dynamic range of a 12-bit ADC.



The bipolar systems offset and gain errors can be calibrated using one of two methods:

- Shift the x and y axes of the transfer function so that the negative full-scale point aligns with the zero point of a unipolar system:

$y = a + (1+b)\ x$

where:

y: digital out;

x: analogue in;

a: offset error;

b: gain error.

❑ Apply zero volts to the ADC input and perform a conversion, in which case the conversion result represents the bipolar zero offset error. Then perform a gain adjustment.

For unipolar systems, the previous method is applicable if the offset is positive. However, if the offset is negative the methodology consists of increasing the input voltage to determine where the first ADC transition occurs.

The gain error can be corrected by software by treating it as a linear function:

$y = (m_1/m_2)\ x$

where,

m_1: slope of the ideal transfer function;

m_2: slope of the measured transfer function.

Both offset and gain errors reduction techniques imply loss of part of the range of the ADC.

❑ ***Other sources of error***

- Code-Edge Noise: Amount of noise that appears right at the code transition of the transfer function;
- Voltage Reference (internal or external): Besides the settling time, the source of reference voltage errors is related to the following specifications:
  - Temperature drift: Affects the performance of an ADC converter based on resolution;
  - Voltage noise: Specified as either an RMS value or a peak-to-peak value;
  - Load regulation: Current drawn by other components will affect the voltage reference;
  - Temperature effects (offset drift and gain drift).

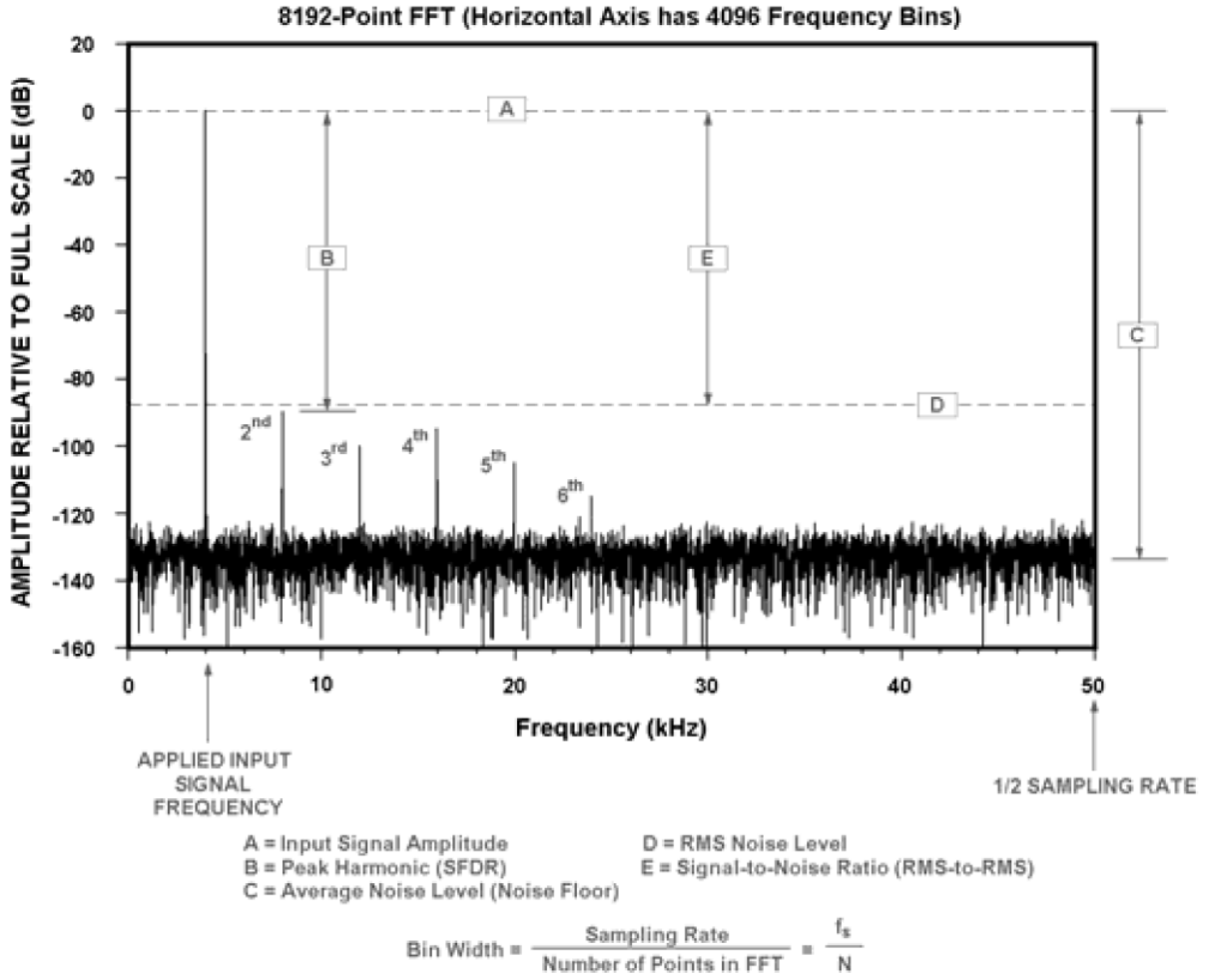
AC Performance

The AC key specifications are:

- Signal-to-noise ratio (SNR);
- Signal-to-noise and distortion ratio (SINAD);
- Total harmonic distortion (THD);
- Spurious-free dynamic range (SFDR).

Harmonics occur at multiples of the input frequency (see *Figure 9-29*).

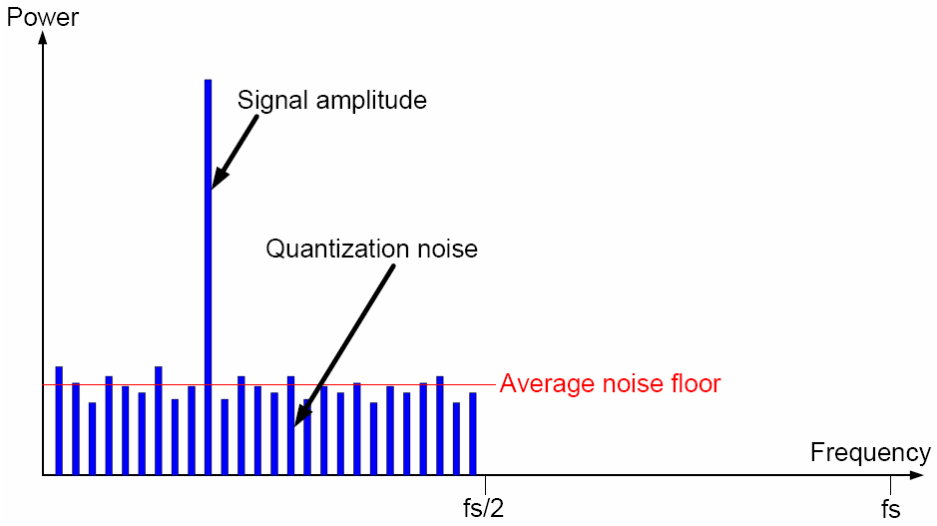
Figure 9-29. AC parameters of a 12-bit ADC.



- ***Signal-to-noise ratio (SNR)***

SNR is the signal-to-noise ratio without distortion components. SNR reveals where the average noise floor of the converter is, and sets the ADC performance limit for noise, as shown in *Figure 9-30*.

Figure 9-30. SNR – average noise floor.

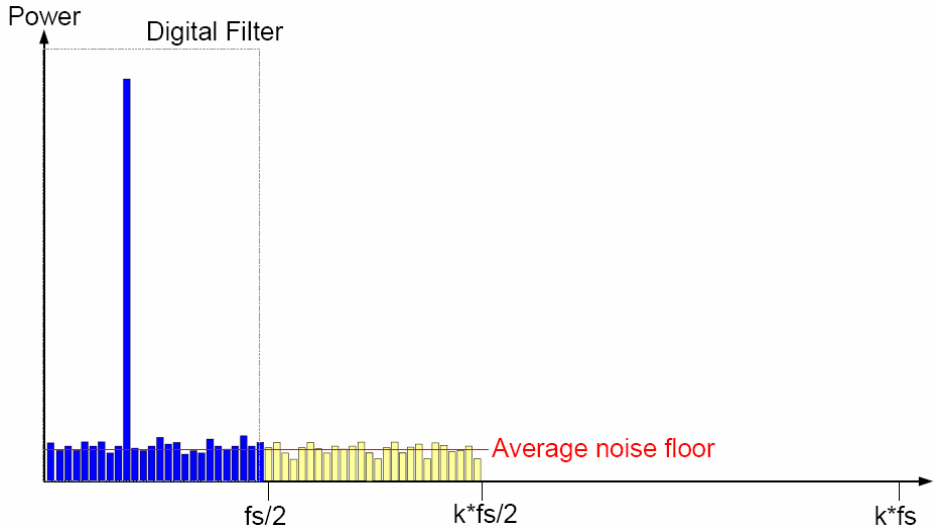


For an *n* bit ADC sine wave input, SNR is given by:

$$SNR = 6.02n + 1.76 \quad [dB]$$

The SNR can be improved by sampling at a rate much higher than the signal of interest (oversampling). This method lowers the average noise floor of the ADC, spreading the noise out over more frequencies, keeping the total noise level the same (see *Figure 9-31*).

Figure 9-31. Oversampling.



Oversampling an ADC is a common principle used to increase resolution. It reduces the noise at any particular frequency. A 2x oversampling reduces the noise floor by 3dB, which corresponds to a ½ bit resolution increase. Oversampling by *k* times provides a SNR given by:

$$SNR = 6.02\,n + 1.76 + 10\log_{10}\left(\frac{f_s}{2\,f_{max}}\right) \quad [dB]$$

- ***Signal-to-noise and distortion ratio (SINAD)***

SINAD is similar to SNR, but it includes the harmonic content [total harmonic distortion], from DC to the Nyquist frequency. Thus, SINAD is defined as the ratio of the RMS value of an input sine wave to the RMS value of the noise of the converter. Writing the equation in terns of *n* provides the number of bits that are obtained as a function of the RMS noise. The equation is the definition for effective number of bits, ENOB:

$$n = (SINAD - 1.76)/6.02$$

- ***Total harmonic distortion (THD)***

As shown in *Figure 9-29*, the THD becomes increasingly worse as the input frequency increases. This is the primary reason for ENOB degradation with frequency, because as frequency increases toward the Nyquist limit, SINAD decreases.

- ***Spurious-free dynamic range (SFDR)***

SFDR is defined as the ratio of the RMS value of an input sine wave to the RMS value of the largest trace observed in the frequency domain using an FFT plot (see *Figure 9-28*). If the distortion component is much larger than the signal of interest, the ADC will not convert small input signals, thus limiting its dynamic range.

Further information concerning ADC fundamentals and applications can found in the TI web page. Amongst them, included in Annex E are some documents that provide a deeper insight into the topic of analogue-to-digital conversion:

- Introduction to MSP430 ADCs <slap115.pdf>
- Understanding Data Converters <slaa013.pdf>
- A Glossary of Analogue-to-Digital Specifications and Performance Characteristics <sbaa147a.pdf>
- Optimized Digital Filtering for the MSP430 <slap108.pdf>
- Efficient MSP430 Code Synthesis for an FIR Filter <slaa357.pdf>
- Working with ADCs, OAs and the MSP430 <slap123.pdf>
- Hands-On: Using MSP430 Embedded Op Amps <slap118.pdf>
- Oversampling the ADC12 for Higher Resolution <slaa323.pdf>
- Hands-on Realizing the MSP430 Signal Chain through ADPCM <slap122.pdf>
- Amplifiers and Bits: An Introduction to Selecting Amplifiers for Data Converters <sloa035b.pdf>
- …

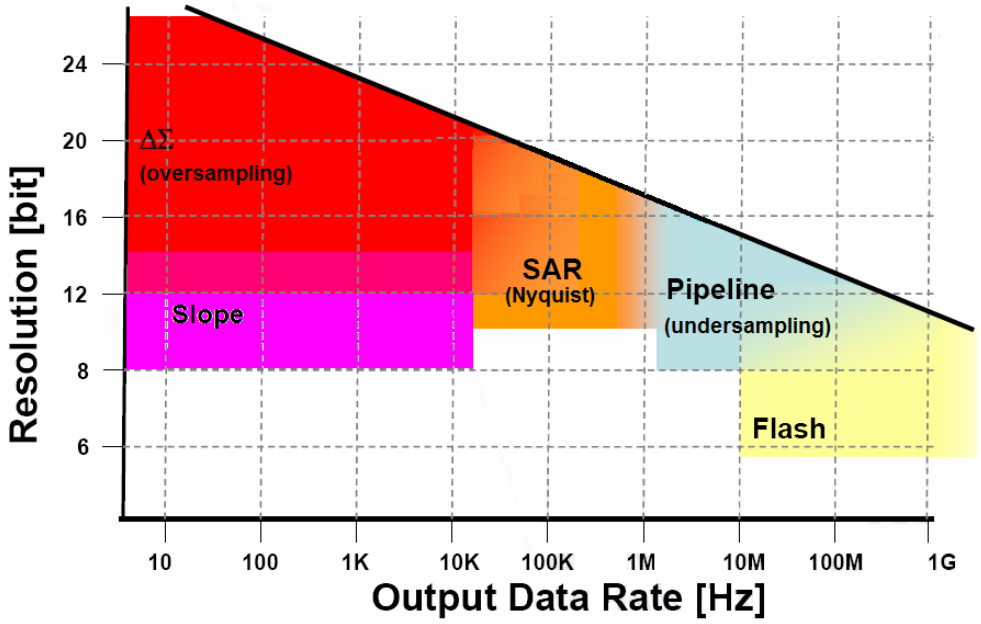
9.3.2 ADC architectures

There are many different ADC architectures:

- Successive Approximation (SAR);
- Sigma Delta (SD or ΔΣ);
- Slope or Dual Slope;
- Pipeline;
- Flash...as in quick, not memory.

Each of these architectures is applicable to a range of output data rates and resolutions as shown in *Figure 9-32*.

Figure 9-32. ADC architectures – Resolution vs. Output data rate.



The selection of an MSP430 ADC will depend on the:

- Voltage range to be measured;
- Maximum frequency for A_{IN};
- Minimum resolution needed to describe the variation in the analogue input;
- Need for differential inputs;
- Reference range;
- Need for multiple channels for different analogue input conversions.

The following table shows the main characteristics for the selection of a particular ADC architecture:

Table 9-6. ADC architecture – characteristics and comments.

ADC architecture	Resolution	Conversion rate	Advantages	Disadvantages
SAR	≤ 18 bit	< 5 Msps	Zero-cycle latency Low latency-time High accuracy Low power Simple operation	Sample rates 2-5 MHz
SD	≤ 24 bit ≤ 16-18 bit	< 625 ksps < 10 Msps	High resolution High stability Low power Moderate cost	Cycle-latency Low speed
Pipeline	≤ 16 bit	< 500 Msps	Higher speeds Higher bandwidth	Lower resolution Delay/Data latency Power requirements

Most MSP430 devices offer a high-precision ADC. Depending on the device, the converter architecture and the resolution can vary. In the MSP430 devices, the following converter architectures are available:

- Slope (Comparator);
- 10-bit SAR;
- 12-bit SAR;
- 16-bit Sigma-Delta.

The ADC architectures included in the MSP430 devices populated in the hardware development tools are as follows:

- 10-bit SAR: MSP430F2274 ⇒ eZ430-RF2500 MSP430 USB Stick Development Tool;
- 12-bit SAR: MSP430FG4618 ⇒ MSP430FG4618/F2013 Experimenter's board;
- 16-bit Sigma-Delta: MSP430F2013 ⇒ eZ430-F2013 MSP430 USB Stick Development Tool and MSP430FG4618/F2013 Experimenter's board.

9.3.3 Successive Approximation Register (SAR) converter

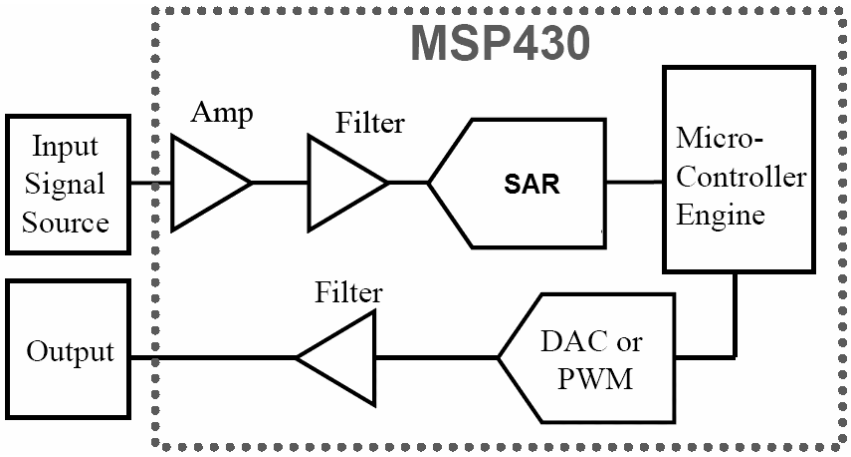
SARs are adequate for general-purpose applications, being widely used in signal level applications:

- Data loggers;
- Temperature sensors;
- Bridge sensors;
- General purpose.

The system integration using a SAR ADC is shown in *Figure 9-33,* taking into account that it usually requires a low-pass filter before the analogue input.

TI offers both stand-alone SARs and integrated SARs in the MSP430. This section will focus the SAR ADCs integrated into the MSP430 devices. Each ADC is a single 10-bit or 12-bit unit, with a built-in sample-and-hold circuit, internal reference and autoscan features. The front end consists of a multiplexer circuit, allowing the developer to select one of eight external pins, or one of four internal sources.

Figure 9-33. System integration using a SAR ADC block diagram.



In the block diagram of a SAR ADC shown in *Figure 9-34*, the method of successive approximation determines the digital word, by approximating the input signal using an iterative process. It uses the following steps:

- Completely discharge the capacitor array to the offset voltage of the comparator;
- Acquire the input voltage (V_S) using the sample and hold;
- Switch all of the capacitors within the array to the input signal, V_S (the capacitors are charged to their respective capacitance x the input voltage - the offset voltage at each of them);
- Switch the capacitors to apply this charge across the comparator's input (comparator input voltage equal to $-V_S$);
- Proceed with the binary search, i.e. with the conversion (see *Figure 9-34* for a 4-bit SAR conversion):
  - Switch the MSB capacitor to V_{REF} (ADC full-scale range):
    - 1:1 divided between it and the rest of the array (due to the binary-weighting of the array);
    - Input voltage to the comparator is $-V_S + V_{REF}/2$;
    - $V_S > V_{REF}/2 \Rightarrow$ Comparator output: MSB = 1;
    - $V_S < V_{REF}/2 \Rightarrow$ Comparator output: MSB = 0;

- Test the other capacitors in a decreasing charge capacity order in the same manner, until the comparator input voltage converges to the offset voltage, or at least as close as possible.

Figure 9-34. SAR ADC block diagram.

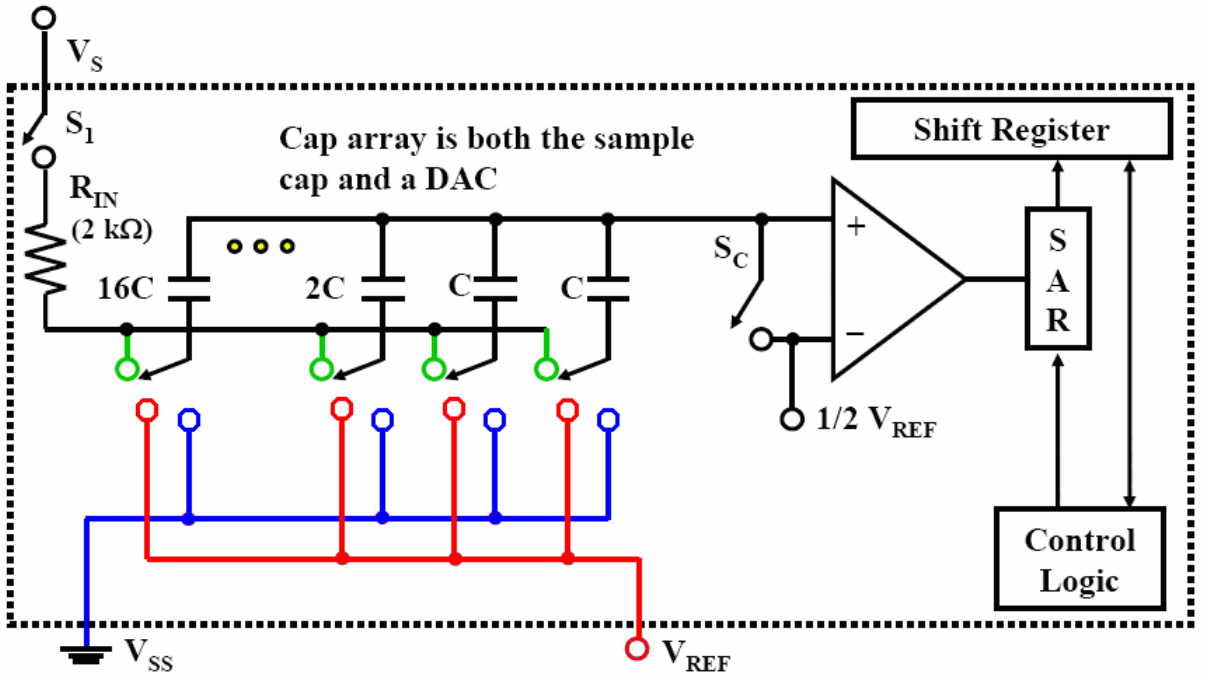
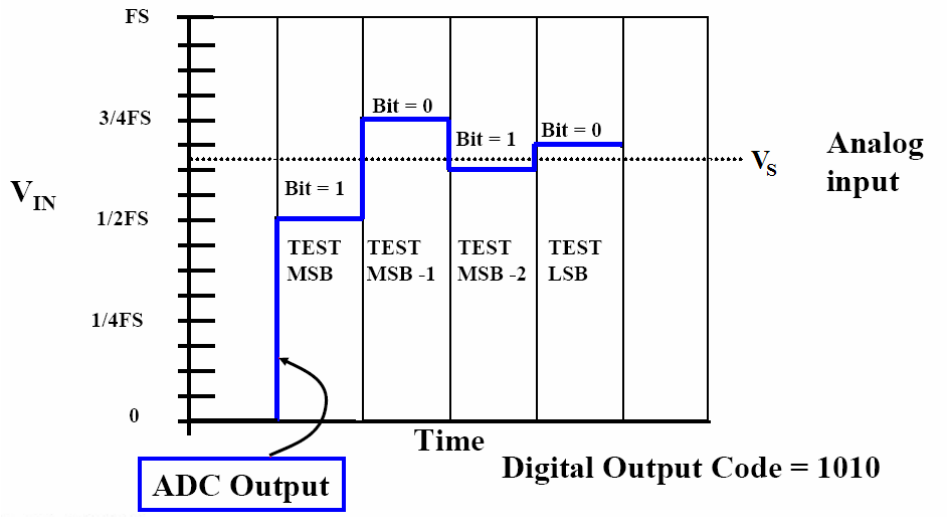


Figure 9-35. SAR analogue-to-digital conversion concept.



ADC10

The ADC10 module included in the MSP430F2274 supports fast, 10-bit analogue-to-digital conversions. The module implements a:

- 10-bit SAR core;
- Sample select control;
- Reference generator;
- Data transfer controller (DTC) for automatic conversion result handling (conversion and storage of ADC samples, without any CPU intervention).

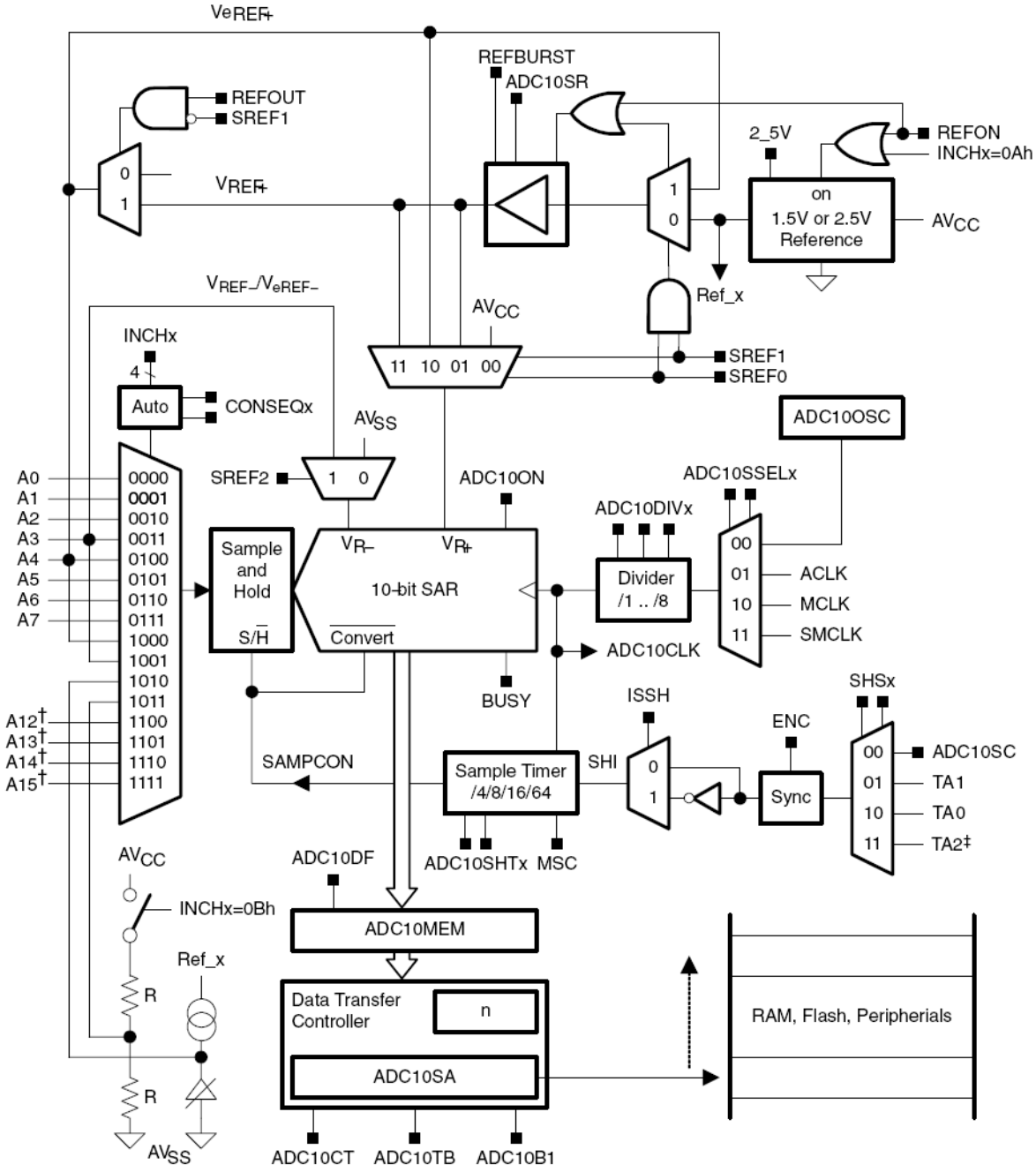
ADC10 features include:

- Greater than 200 ksps maximum conversion rate;
- Monotonic 10-bit converter with no missing codes;
- Sample-and-hold with programmable sample periods;
- Conversion initiation by software or Timer_A;
- Software selectable on-chip reference voltage generation (1.5 V or 2.5 V)
- Software selectable internal or external reference;
- Eight external input channels;
- Conversion channels for internal temperature sensor, V_{CC}, and external references;
- Selectable conversion clock source;
- Single-channel, repeated single-channel, sequence, and repeated sequence conversion modes;
- ADC core and reference voltage can be powered down separately;
- Data transfer controller for automatic storage of conversion results.

The ADC10 module is configured by user software. Its setup and operation are discussed in the following sections.

The block diagram of the ADC10 is shown in *Figure 9-36*.

Figure 9-36. SAR ADC10 block diagram.



†MSP430x22xx devices only. Channels A12-A15 tied to channel A11 in other devices
‡TA1 on MSP430x20x2 devices

❑ *10-bit ADC core*

The ADC core converts an analogue input to its 10-bit digital representation and stores the result in the ADC10MEM register.

The analogue conversion range is limited to its upper and lower limits (V_{R+} and V_{R-}), programmable/selectable voltage.

The digital output (N_{ADC}) is:

- Full scale: N_{ADC} = 03FFh, when the input signal $\geq V_{R+}$ - 0.5LSB;

- Zero: N_{ADC} = 0000h, when the input signal ≤ V_{R-} + 0.5LSB.

Conversion results may be in binary format or Two's-complement format. Using binary format, the conversion formula for the ADC result is given by:

$$N_{ADC} = 1023 \frac{V_{in} - V_{R-}}{V_{R+} - V_{R-}}$$

The ADC10 core is enabled with the ADC10ON bit. Then, it can be configured by two control registers, ADC10CTL0 and ADC10CTL1. These control bits can only be modified when ENC = 0, because this bit must be set to start the conversions.

When it is not actively converting, the core is automatically disabled and automatically re-enabled when required.

- ***Conversion clock selection***

The ADC10CLK is used both as the conversion clock and to generate the sampling period.

Each available ADC10 source clock is selected using the ADC10SSELx bits:

- SMCLK;
- MCLK;
- ACLK;
- Internal oscillator ADC10OSC;
- Each one can be divided by 1-8 using the ADC10DIVx bits.

The ADC10CLK must remain active until the end of a conversion. If the clock is removed during a conversion, the operation will not complete, and any result will be invalid. However, the ADC10OSC is also automatically enabled when required and disabled when not required during conversions.

- ***ADC10 inputs and multiplexer***

The analogue input multiplexer allows selection of one of the eight external and four internal analogue signals, as the channel for conversion.

- ***Analogue port selection***

The external inputs Ax, Ve_{REF+}, and V_{REF-} share terminals with GPIO pins. To reduce parasitic current surges between V_{CC} and GND and consequently higher current consumption, the input/output port pin buffer can be disabled from the port pin buffer by setting the ADC10AEx bits.

❑ *Voltage reference generator*

The internal voltage reference (1.5 V or 2.5 V) can be selected by disabling or enabling the REFON bit, respectively. This internal reference generator is designed for low power applications, consisting of a band-gap voltage source and a separate buffer. The buffer will be enabled or disabled, depending on the conversion status, being active or inactive, respectively.

The internal reference buffer has selectable speed vs. power settings (example: $f_{conversion} \leq 50$ ksps and ADC10SR = 1 $\Rightarrow$ $I_{buffer} \approx 50\%$ reduction). The total settling time when REFON = 1 is ≈ 30 µs.

The internal reference voltage can also be used externally (pin V_{REF+}), by disabling the REFOUT bit. External references may supply V_{R+} and V_{R-} or use V_{CC} as reference. In these cases, the internal reference may be turned off to save power. To use an external positive reference Ve_{REF+}, the SREFx bits must be set.

❑ *Sample and conversion timing*

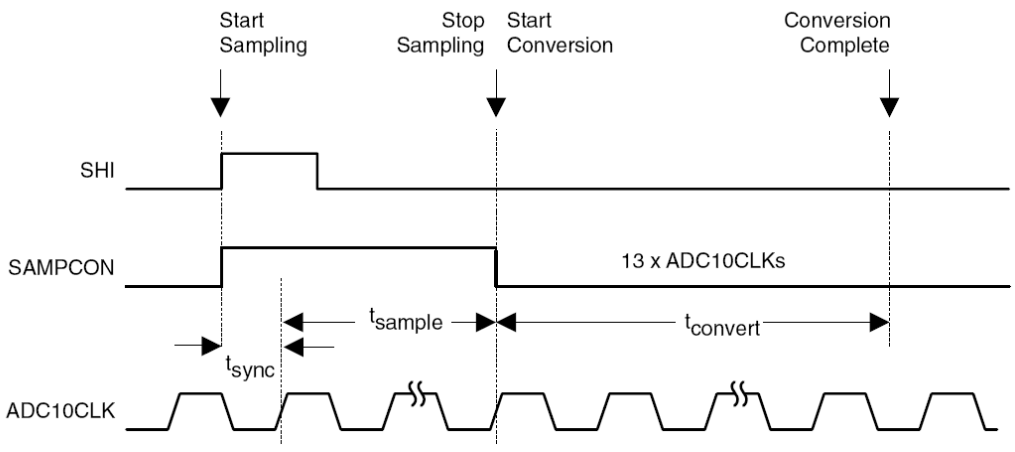
An analogue-to-digital conversion is initiated on a rising edge of sample input signal SHI. The source for SHI (SHSx bits selection) can be the:

- ❑ ADC10SC bit;
- ❑ Timer_A Output Unit 1;
- ❑ Timer_A Output Unit 0;
- ❑ Timer_A Output Unit 2.

The polarity of the SHI signal source can be inverted with the ISSH bit.

The SHTx bits select the sample period, t_{sample}, to be 4, 8, 16, or 64 ADC10CLK cycles. The sampling timer sets SAMPCON high for the selected sample period after synchronization with ADC10CLK. So, the total sampling time (see *Figure 9-37*) is: $t_{tot} = t_{sample} + t_{sync}$

Figure 9-37. SAR ADC10 sample timing.



- ***Conversion modes***

The ADC10 has four operating modes selected by the CONSEQx bits:

- Single channel, single-conversion: One single conversion for the channel selected by INCHx bits, with the result being stored in the ADC10MEM registers;
- Sequence of channels: One conversion in multiple channels beginning with the channel selected by INCHx bits and decrements to channel A0, looping through a specified number of ADC10MEM registers and stopping after the conversion of channel A0;
- Repeat single channel: A single channel selected by INCHx bits is converted repeatedly until stopped, storing the result in the ADC10MEM register;
- Repeat sequence of channels: Repeated conversions through multiple channels, beginning with the channel selected by INCHx bits and decrementing to channel A0. Each ADC result is written to ADC10MEM. The sequence ends after conversion of channel A0, and the next trigger signal re-starts the sequence.

- ***Re-start conversion***

Successive conversions can start automatically and very quickly when MSC = 1 and CONSEQx > 0. The first rising edge of the SHI signal triggers the first conversion and the next conversions are triggered automatically as soon as the previous conversion is completed, without needing additional rising edges on SHI.

- ***Stopping conversions***

Stopping conversions depends on the mode of operation:

- Single-channel single-conversion mode: Poll the ADC10BUSY bit until reset and than reset ENC bit;
- Repeat-single-channel mode: Reset ENC bit;
- Sequence or repeat sequence mode: Reset ENC.
- Any conversion mode may be stopped immediately by resetting CONSEQx bits and ENC bit. However, the conversion data is unreliable.

- ***Data Transfer Controller (DTC)***

The DTC (ADC10DTC1 $\neq$ 0) automatically transfers the conversion results from ADC10MEM to other on-chip memory locations, each time the ADC10 completes a conversion and loads the result to ADC10MEM. Since it requires one CPU MCLK, if the CPU is active during this period, it will be halted to ensure the transfer is completed. Additionally, it must be ensured that no active conversion or sequence is in progress (ADC10 busy) during the DTC transfer initiation.

The DTC can be configured for:

❑ One-Block Transfer Mode (ADC10TB = 0): The value *n* in ADC10DTC1 defines the total number of transfers for a block. The block start address by the 16-bit register ADC10SA and it ends at ADC10SA+2n–2.

❑ Two-Block Transfer Mode (ADC10TB = 1): The value *n* in ADC10DTC1 defines the number of transfers for one block. The address range of the first block is defined by the 16-bit register ADC10SA and it ends at ADC10SA+2n–2. The address range for the second block is defined as SA+2n to SA+4n–2.

A continuous transfer (ADC10CT = 1) indicates that the DTC will not stop after block one in (one-block mode) or block two (two-block mode) has been transferred. Transfers continue in block one.

❑ *Integrated temperature sensor*

The analogue input channel INCHx = 1010 uses the on-chip temperature sensor. Its transfer function relating the input voltage, $V_{\text{Temperature}}$ [*V*] to the temperature, *T* [ºC], is given by:

$$V_{Temperature} = 0.00355 \times T + 0.986$$

When using the temperature sensor, the following should be taken into consideration:

❑ The sample period must be greater than 30 µs;

❑ As the offset error is large, the applications must include calibration;

❑ Selecting INCHx = 1010 automatically turns on the on-chip reference generator as a voltage source;

❑ The reference choices for converting the temperature obtained with the integrated sensor are the same as with any other channel.

❑ *ADC10 interrupts*

One interrupt and one interrupt vector are associated with ADC10.

❑ When the DTC is not used (ADC10DTC1 = 0): ADC10IFG is set when conversion results are loaded into ADC10MEM.

❑ When the DTC is used (ADC10DTC1 > 0): ADC10IFG is set when a block transfer completes and the internal transfer counter *n* = 0.

If both the ADC10IE and the GIE bits are set, then the ADC10IFG flag generates an interrupt request. The ADC10IFG flag is automatically reset when the interrupt request is serviced or may be reset by software.

❑ *ADC10 registers*

ADC10CTL0, ADC10 Control Register 0

15	14	13	12	11	10	9	8
SREFx			ADC10SHTx		ADC10SR	REFOUT	REFBURST

7	6	5	4	3	2	1	0
MSC	REF2_5V	REFON	ADC10ON	ADC10IE	ADC10IFG	ENC	ADC10SC

ADC10CTL0, ADC10 Control Register 0 (high byte)

Bit		Description		
15-13	SREFx	Select voltage reference:	**V_{R+}**	**V_{R-}**
		SREF2 SREF1 SREF0 = 000 ⇒	V_{CC}	V_{SS}
		SREF2 SREF1 SREF0 = 001 ⇒	V_{REF+}	V_{SS}
		SREF2 SREF1 SREF0 = 010 ⇒	Ve_{REF+}	V_{SS}
		SREF2 SREF1 SREF0 = 011 ⇒	Buffered Ve_{REF+}	V_{SS}
		SREF2 SREF1 SREF0 = 100 ⇒	V_{CC}	V_{REF-}/Ve_{REF-}
		SREF2 SREF1 SREF0 = 101 ⇒	V_{REF+}	V_{REF-}/Ve_{REF-}
		SREF2 SREF1 SREF0 = 110 ⇒	Ve_{REF+}	V_{REF-}/Ve_{REF-}
		SREF2 SREF1 SREF0 = 111 ⇒	Buffered Ve_{REF+}	V_{REF-}/Ve_{REF-}
12-11	ADC10SHTx	ADC10 sample-and-hold time: ADC10SHT1 ADC10SHT0 = 00 ⇒ 4 x ADC10CLKs ADC10SHT1 ADC10SHT0 = 01 ⇒ 8 x ADC10CLKs ADC10SHT1 ADC10SHT0 = 10 ⇒16 x ADC10CLKs ADC10SHT1 ADC10SHT0 = 11 ⇒ 64 x ADC10CLKs		
10	ADC10SR	ADC10 sampling rate: ADC10SR = 0 ⇒ Reference buffer supports up to ~200 ksps ADC10SR = 1 ⇒ Reference buffer supports up to ~50 ksps		
9	REFOUT	Reference voltage output (pin V_{REF+}): REFOUT = 0 ⇒ Disable REFOUT = 1 ⇒ Enable		
8	REFBURST	Controls the operation of the internal reference buffer: REFBURST = 0 ⇒ Reference buffer on continuously allowing the reference voltage to be present outside the device continuously. REFBURST = 1 ⇒ Reference buffer automatically disabled when the ADC10 is not actively converting, and automatically re-enabled when during sample-and-conversion.		

ADC10CTL0, ADC10 Control Register 0 (low byte)

Bit		Description
7	MSC	Multiple sample and conversion (Valid for sequence or repeated modes): MSC = 0 ⇒ Requires a rising edge of the SHI signal to trigger each sample-and-conversion. MSC = 1 ⇒ After the first rising edge of the SHI signal that triggers the sampling timer the further sample-and-conversions are performed automatically as soon as the prior conversion is completed
6	REF2_5V	Reference-generator voltage select (REFON bit must also be set): REF2_5V = 0 ⇒ Reference voltage = 1.5 V REF2_5V = 1 ⇒ Reference voltage = 2.5 V
5	REFON	Reference generator: REFON = 0 ⇒ Reference generator disable REFON = 1 ⇒ Reference generator enable
4	ADC10ON	ADC10 on: ADC10ON = 0 ⇒ ADC10 off ADC10ON = 1 ⇒ ADC10 on
3	ADC10IE	ADC10 interrupt enable ADC10IE = 0 ⇒ Interrupt disabled ADC10IE = 1 ⇒ Interrupt enabled
2	ADC10IFG	ADC10 interrupt flag: ADC10IFG = 0 ⇒ No interrupt pending (interrupt request is accepted, or it may be reset by software) ADC10IFG = 1 ⇒ Interrupt pending (ADC10MEM is loaded with a conversion result or when a block of DTC transfers is completed)
1	ENC	Enable conversion: ENC = 0 ⇒ ADC10 disabled ENC = 1 ⇒ ADC10 enabled
0	ADC10SC	Start conversion: ADC10SC = 0 ⇒ No sample-and-conversion start ADC10SC = 1 ⇒ Start sample-and-conversion

ADC10CTL1, ADC10 Control Register 1

15	14	13	12	11	10	9	8
INCHx				SHSx		ADC10DF	ISSH

7	6	5	4	3	2	1	0
ADC10DIVx			ADC10SSELx		CONSEQx		ADC10BUSY

Bit		Description
15–12	INCHx	Input channel select: INCH3 INCH2 INCH1 INCH0 = 0000 ⇒ A0 INCH3 INCH2 INCH1 INCH0 = 0001 ⇒ A1 INCH3 INCH2 INCH1 INCH0 = 0010 ⇒ A2 INCH3 INCH2 INCH1 INCH0 = 0011 ⇒ A3 INCH3 INCH2 INCH1 INCH0 = 0100 ⇒ A4 INCH3 INCH2 INCH1 INCH0 = 0101 ⇒ A5 INCH3 INCH2 INCH1 INCH0 = 0110 ⇒ A6 INCH3 INCH2 INCH1 INCH0 = 0111 ⇒ A7 INCH3 INCH2 INCH1 INCH0 = 1000 ⇒ Ve_{REF+} INCH3 INCH2 INCH1 INCH0 = 1001 ⇒ V_{REF-}/Ve_{REF-} INCH3 INCH2 INCH1 INCH0 = 1010 ⇒ Temperature sensor INCH3 INCH2 INCH1 INCH0 = 1011 ⇒ $(V_{CC} - V_{SS})/2$ INCH3 INCH2 INCH1 INCH0 = 1100 ⇒ $(V_{CC} - V_{SS})/2$ or A12* INCH3 INCH2 INCH1 INCH0 = 1101 ⇒ $(V_{CC} - V_{SS})/2$ or A13 * INCH3 INCH2 INCH1 INCH0 = 1110 ⇒ $(V_{CC} - V_{SS})/2$ or A14 * INCH3 INCH2 INCH1 INCH0 = 1111 ⇒ $(V_{CC} - V_{SS})/2$ or A15 * * on MSP430x22xx devices
11–10	SHSx	Sample-and-hold source: SHS1 SHS0 = 00 ⇒ bit ADC10SC SHS1 SHS0 = 01 ⇒ TIMER_A Output Unit 1 SHS1 SHS0 = 10 ⇒ TIMER_A Output Unit 0 SHS1 SHS0 = 11 ⇒ TIMER_A Output Unit 2
9	ADC10DF	ADC10 data format: ADC10DF = 0 ⇒ Binary ADC10DF = 1 ⇒ Two's complement
8	ISSH	Invert signal sample-and-hold ISSH = 0 ⇒ The sample-input signal is not inverted ISSH = 1 ⇒ The sample-input signal is inverted
7–5	ADC10DIVx	ADC10 clock divider: ADC10DIV2 ADC10DIV1 ADC10DIV0 = 000 ⇒ / 1 ADC10DIV2 ADC10DIV1 ADC10DIV0 = 001 ⇒ / 2 ADC10DIV2 ADC10DIV1 ADC10DIV0 = 010 ⇒ / 3 ADC10DIV2 ADC10DIV1 ADC10DIV0 = 011 ⇒ / 4 ADC10DIV2 ADC10DIV1 ADC10DIV0 = 100 ⇒ / 5 ADC10DIV2 ADC10DIV1 ADC10DIV0 = 101 ⇒ / 6 ADC10DIV2 ADC10DIV1 ADC10DIV0 = 110 ⇒ / 7 ADC10DIV2 ADC10DIV1 ADC10DIV0 = 111 ⇒ / 8
4–3	ADC10SSELx	ADC10 clock source: ADC10SSEL1 ADC10SSEL0 = 00 ⇒ ADC10OSC ADC10SSEL1 ADC10SSEL0 = 01 ⇒ ACLK ADC10SSEL1 ADC10SSEL0 = 10 ⇒ MCLK ADC10SSEL1 ADC10SSEL0 = 11 ⇒ SMCLK
2–1	CONSEQx	Conversion sequence mode: CONSEQ1 CONSEQ0 = 00 ⇒ Single-channel, single-conversion CONSEQ1 CONSEQ0 = 01 ⇒ Sequence-of-channels CONSEQ1 CONSEQ0 = 10 ⇒ Repeat-single-channel CONSEQ1 CONSEQ0 = 11 ⇒ Repeat-sequence-of-channel
0	ADC10BUSY	ADC10 busy: ADC10BUSY = 0 ⇒ No operation is active ADC10BUSY = 1 ⇒ Sequence, sample, or conversion is active

ADC10AE0, Analogue (Input) Enable Control Register 0

Setting the bits of this 8-bit register enables the analogue input of the ADC10, where BIT0 corresponds to A0, BIT1 corresponds to A1, and so on.

ADC10AE1, Analogue (Input) Enable Control Register 1 (MSP430x22xx only)

Devices such as the MSP430F2274, which is included in the eZ430-RF2500, contain an 8-bit (4 most significant bits) additional analogue input enable control register. BIT4 corresponds to A12, BIT5 corresponds to A13, BIT6 corresponds to A14, and BIT7 corresponds to A15.

ADC10MEM, Conversion-Memory Register

These 16-bit registers are loaded with the conversion results. The number of available bits depends on the numerical result format:

Binary: Bits 15-10 are always 0. The conversion results are stored in the least significant 10 bits, Bit 9 being the MSB.

Two's complement: The conversion results are stored in the most significant 10 bits, Bit 15 being the MSB. Bits 5-0 are always 0.

ADC10DTC0, Data Transfer Control Register 0

7	6	5	4	3	2	1	0
Reserved				ADC10TB	ADC10CT	ADC10B1	ADC10FETCH

Bit		Description
3	ADC10TB	ADC10 block mode: ADC10TB = 0 ⇒ One-block transfer mode ADC10TB = 1 ⇒ Two-block transfer mode
2	ADC10CT	ADC10 continuous transfer ADC10CT = 0 ⇒ Data transfer stops when a block(s) transfer is completed ADC10CT = 1 ⇒ Data is transferred continuously
1	ADC10B1	block filled with ADC10 conversion results (two-block mode): ADC10B1 = 0 ⇒ Block 2 is filled ADC10B1 = 1 ⇒ Block 1 is filled
0	ADC10FETCH	Normally set ADC10FETCH = 0

ADC10DTC1, Data Transfer Control Register 1

This 8-bit register defines the number of transfers in each block.

ADC10DTC1 = 0 ⇒ DTC is disabled

ADC10DTC1 = 01h−0FFh ⇒ Number of transfers per block

ADC10SA, Start Address Register for Data Transfer

This 16-bit register defines the ADC10 start address for the DTC. It uses only the 15 most significant bits. Bit 0 is always read as 0.

ADC12

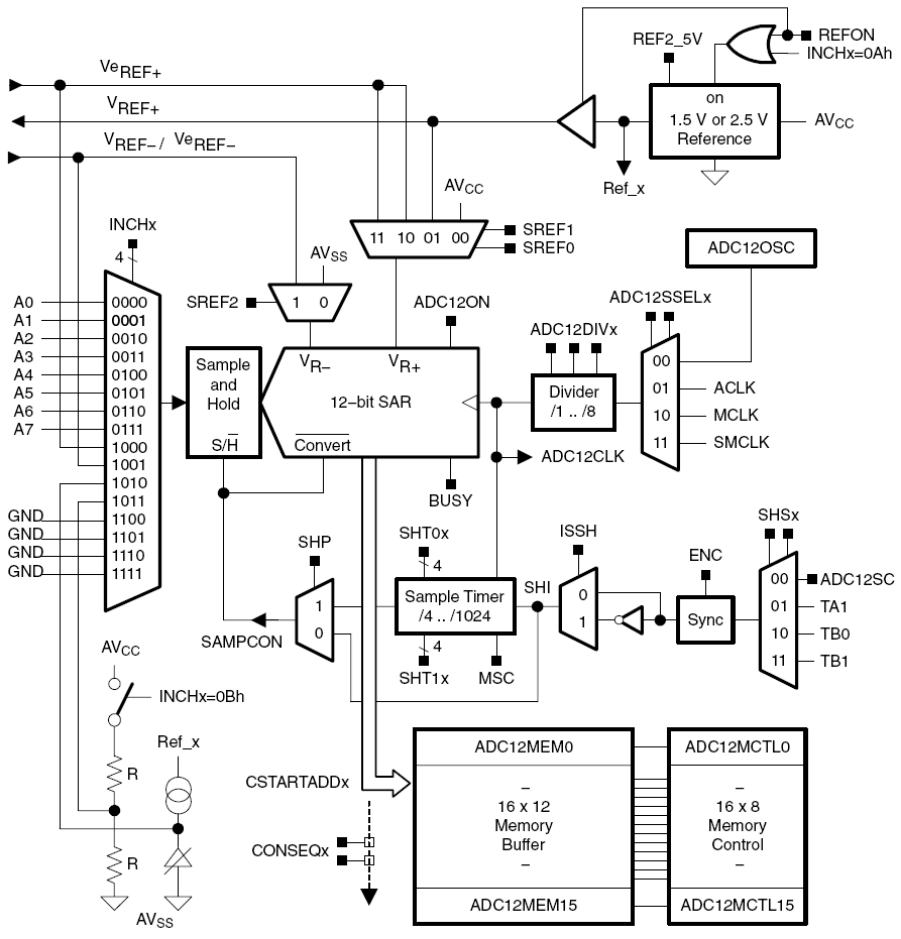
The ADC12 module, which is included in the MSP430FG4618, supports fast 12-bit analogue-to-digital conversions. The module contains:

- 12-bit SAR core;
- Sample select control;
- Reference generator;

It has same basic features as ADC10, the differences being:

- Monotonic 12-bit converter with no missing codes;
- Interrupt vector register for fast decoding of 18 ADC interrupts;
- 16 conversion result storage registers;
- No Data Transfer Controller (DTC);
- 16 control registers ADC12MCTLx free choice of channels on sequential modes;
- Also some channels more than once in one loop (e.g. placing two conversion of the same voltage and one measurement of current in the middle to calculate power).

Figure 9-38. SAR ADC12 block diagram.



The ADC12 follows the same design philosophy as the ADC10, however it does have some differences that will be described in the following sections:

- ***12 bit ADC core***

As with the ADC10, the conversion is limited by the upper and lower limits (V_{R+} and V_{R-}), which are programmable/selectable voltages.

The digital output (N_{ADC}) is:

- Full scale: N_{ADC} = 0FFFh, when the input signal $\geq V_{R+}$;
- Zero: N_{ADC} = 0000h, when the input signal $\leq V_{R-}$.

Conversion results may be in binary format or Two's-complement format. Using binary format, the conversion equation for the ADC result is given by:

$$N_{ADC} = 4096 \frac{V_{in} - V_{R-}}{V_{R+} - V_{R-}}$$

- ***Conversion clock selection***

The ADC12CLK is similar to the ADC10CLK.

- ***ADC12 inputs and multiplexer***

The ADC12 inputs and multiplexer have the same structure as the ADC10.

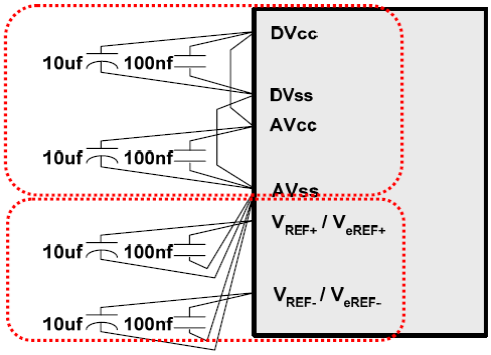
- ***Analogue port selection***

The ADC12 inputs are multiplexed with the port P6 pins. The P6SELx bits provide the ability to disable the port pin input and output buffers, in order to eliminate the parasitic current flow and therefore reduce overall current consumption.

Voltage reference generator

The ADC12 provides the same internal voltage reference generator as the ADC10. For proper operation of the internal voltage reference generator, the ADC12 requires storage capacitance across V_{REF+} and AV_{SS} (recommended parallel combination of 10 µF and 0.1 µF capacitors).

Figure 9-39. SAR ADC12 Reference Decoupling.



With this configuration, a 17 msec time period must be allowed for the voltage reference generator to apply bias to the recommended storage capacitors.

- ***Sample and conversion timing***

As with the ADC10, analogue-to-digital conversion in the ADC12 is initiated on a rising edge at the sample input signal SHI, but the source for SHI (SHSx bits selection) is selected by:

- ADC12SC bit;
- Timer_A Output Unit 1;
- Timer_B Output Unit 0;
- Timer_B Output Unit 1.

The automatic Timer_A/Timer_B SOC triggers are used to eliminate phase error, improving the accuracy and low power consumption.

The timer trigger should be used for Ultra Low Power (ULP) periodic measurements, following the procedure shown in *Figure 9-40,* with the time periods shown in *Figure 9-41*.

Figure 9-40. SAR ADC12 ULP procedure.

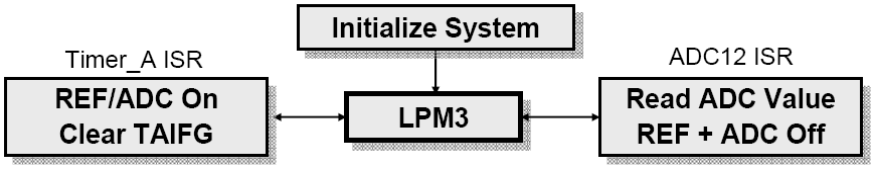
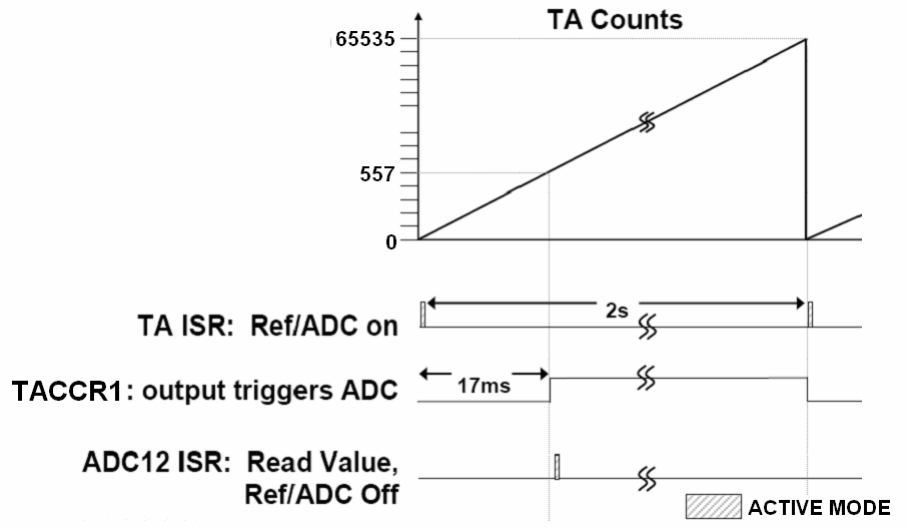


Figure 9-41. ADC12 Timer trigger for reference settling.



The SAMPCON signal controls the sample period and start of conversion:

- SAMPCON = 1 ⇒ sampling is active;
- High-to-Low SAMPCON transition ⇒ starts the analogue-to-digital conversion (13 ADC12CLK cycles).
- Two different methods of sample timing can be defined (SHP bit):
  - SHP = 0: Extended sample mode (see *Figure 9-42*):
    - SHI signal directly controls SAMPCON;
    - Defines the length of the sample period t_{sample};
    - SAMPCON = 1 ⇒ sampling is active;
    - High-to-Low SAMPCON transition ⇒ starts the conversion after synchronization with ADC12CLK.
  - SHP = 1: Pulse mode (see *Figure 9-43*):
    - SHI signal triggers the sampling timer;
    - SHT0x and SHT1x bits (ADC12CTL0) defines the SAMPCON sample period, t_{sample};
    - The sampling timer keeps SAMPCON = 1 after synchronization with ADC12CLK;
    - $t_{tot} = t_{sample} + t_{sync}$;
    - SHTx bits select the sampling time in 4x multiples of ADC12CLK. SHT0x sets the sampling time for ADC12MCTL0 to 7 and SHT1x sets the sampling time for ADC12MCTL8 to 15.

Figure 9-42. SAR ADC12 – Extended sample mode.

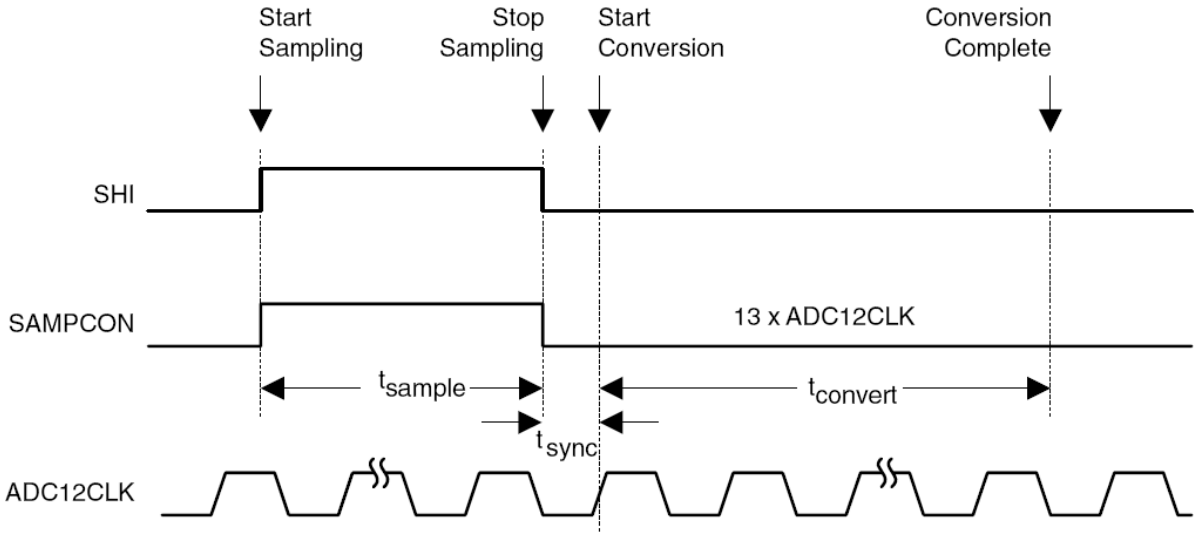
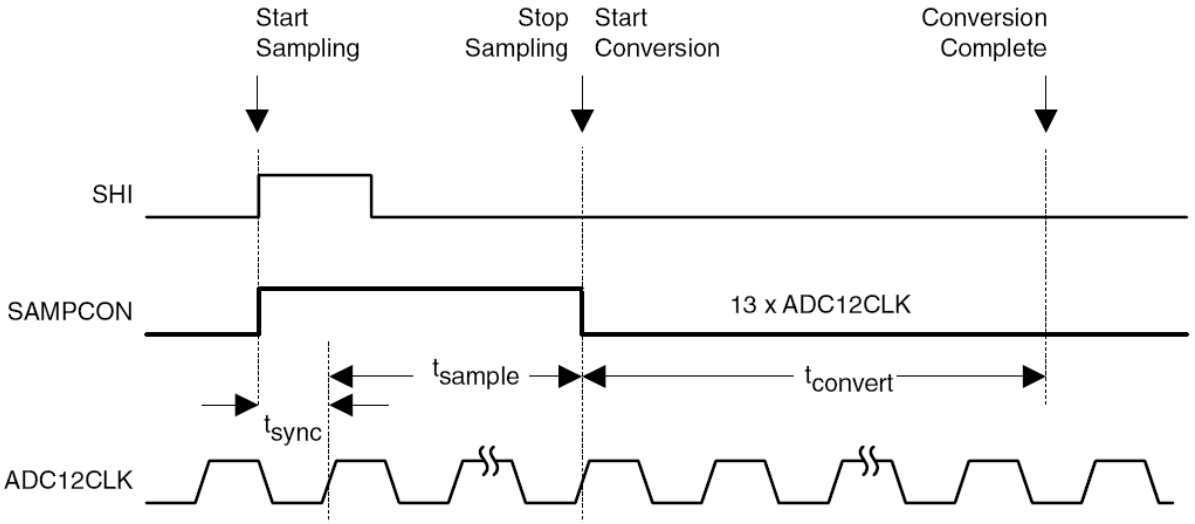


Figure 9-43. SAR ADC12 – Pulse sample mode.



❑ ***Conversion modes***

The ADC12 has the same four operating modes as ADC10, which are selected by the CONSEQx bits. Refer to the ADC10 conversion modes section, taking into account that the register nomenclature is slightly different. For more detailed register definition see the ADC12 registers section.

❑ ***Conversion memory***

The ADC12 has 16 ADC12MEMx conversion memory registers (configured with an associated ADC12MCTLx control register) to store conversion results.

- Non-sequential conversion (single-channel or repeat-single-channel):
  - CSTARTADDx bits define the first and single ADC12MCTLx used for any conversion.
- Sequential conversion (sequence-of-channels or repeat-sequence-of-channels):
  - EOS bit defines the end of sequence;
  - A sequence rolls over from ADC12MEM15 to ADC12MEM0 when the EOS bit is reset;
  - CSTARTADDx points to the first ADC12MCTLx location to be used in a sequence;
  - A pointer is incremented automatically to the next ADC12MCTLx in a sequence when each conversion completes;

For both conversion types, when conversion results are written to the selected ADC12MEMx, the corresponding flag in the ADC12IFGx register is set.

Re-start conversion

Successive conversions are defined as for the ADC10.

Stopping conversions

Stopping conversions also depends on the mode of operation, but is similar to the ADC10.

Integrated temperature sensor

As per the ADC10, the analogue input channel INCHx = 1010 uses the on-chip temperature sensor. Refer to the ADC10 integrated temperature sensor for additional details.

ADC12 interrupts

The ADC12 has 18 interrupt sources:

- ADC12IFG0-ADC12IFG15: ADC12IFGx bits are set when their corresponding ADC12MEMx memory register is loaded with a conversion result;
- ADC12OV, ADC12MEMx overflow: ADC12OV is set when a conversion result is written to any ADC12MEMx, before its previous conversion result was read;

❑ ADC12TOV, ADC12 conversion time overflow: ADC12TOV is set when another sample-and-conversion is requested before the current conversion is completed;

An interrupt request is generated if the corresponding ADC12IEx bit and the GIE bit are set.

The DMA is triggered after the conversion in single channel modes or after the completion of sequence−of−channel modes.

❑ *Interrupt vector generator (ADC12IV)*

The interrupt vector register ADC12IV is used to determine which enabled ADC12 interrupt source requested an interrupt. The following conditions are taken into consideration by the ADC12 interrupt handler:

❑ The highest priority enabled ADC12 interrupt generates a number in the ADC12IV register that can be evaluated or added to the program counter to automatically enter the appropriate software routine. Disabled ADC12 interrupts do not affect the ADC12IV value;

❑ Any access, read or write, of the ADC12IV register automatically resets the ADC12OV condition or the ADC12TOV condition if either was the highest pending interrupt. Neither interrupt condition has an accessible interrupt flag;

❑ The ADC12IFGx flags are not reset by an ADC12IV access. ADC12IFGx bits are reset automatically by accessing their associated ADC12MEMx register or may be reset with software;

❑ If another interrupt is pending after servicing of an interrupt, another interrupt is generated.

❑ *ADC12 registers*

ADC12CTL0, ADC12 Control Register 0

15	14	13	12	11	10	9	8
SHT1x				SHT0x			

7	6	5	4	3	2	1	0
MSC	**REF2_5V**	**REFON**	**ADC12ON**	ADC12OVIE	ADC12TOVIE	**ENC**	**ADC12SC**

The **bold** bits have the same function as the ADC10. Refer to the ADC10 section for their description. The following table highlights only the cases where the description of the different bits differs from the ADC10.

Bit		Description
15-12	SHT1x	Sample-and-hold time (ADC12CLK cycles in the sampling period for registers ADC12MEM8 to ADC12MEM15): SHT13 SHT12 SHT11 SHT10 = 0000 ⇒ 4 ADC12CLK cycles SHT13 SHT12 SHT11 SHT10 = 0001 ⇒ 8 ADC12CLK cycles SHT13 SHT12 SHT11 SHT10 = 0010 ⇒ 16 ADC12CLK cycles SHT13 SHT12 SHT11 SHT10 = 0011 ⇒ 32 ADC12CLK cycles SHT13 SHT12 SHT11 SHT10 = 0100 ⇒ 64 ADC12CLK cycles SHT13 SHT12 SHT11 SHT10 = 0101 ⇒ 96 ADC12CLK cycles SHT13 SHT12 SHT11 SHT10 = 0110 ⇒ 128 ADC12CLK cycles SHT13 SHT12 SHT11 SHT10 = 0111 ⇒ 192 ADC12CLK cycles SHT13 SHT12 SHT11 SHT10 = 1000 ⇒ 256 ADC12CLK cycles SHT13 SHT12 SHT11 SHT10 = 1001 ⇒ 384 ADC12CLK cycles SHT13 SHT12 SHT11 SHT10 = 1010 ⇒ 512 ADC12CLK cycles SHT13 SHT12 SHT11 SHT10 = 1011 ⇒ 768 ADC12CLK cycles SHT13 SHT12 SHT11 SHT10 = 1100 ⇒ 1024 ADC12CLK cycles SHT13 SHT12 SHT11 SHT10 = ⇒ 1101 1024 ADC12CLK cycles SHT13 SHT12 SHT11 SHT10 = ⇒ 1110 1024 ADC12CLK cycles SHT13 SHT12 SHT11 SHT10 = ⇒ 1111 1024 ADC12CLK cycles
11-8	SHT0x	Sample-and-hold time (ADC12CLK cycles in the sampling period for registers ADC12MEM0 to ADC12MEM7). These bits are configured as the previous ones (SHT1x).
3	ADC12OVIE	ADC12MEMx overflow-interrupt enable (The GIE bit must also be set to enable the interrupt): ADC12OVIE = 0 ⇒ Overflow interrupt disabled ADC12OVIE = 1 ⇒ Overflow interrupt enabled
2	ADC12TOVIE	ADC12 conversion-time-overflow interrupt enable (The GIE bit must also be set to enable the interrupt): ADC12TOVIE = 0 ⇒ Conversion time overflow interrupt disabled ADC12TOVIE = 1 ⇒ Conversion time overflow interrupt enabled

ADC12CTL1, ADC12 Control Register 1

15	14	13	12	11	10	9	8
CSTARTADDx				**SHSx**		SHP	**ISSH**

7	6	5	4	3	2	1	0
ADC12DIVx			**ADC12SSELx**		**CONSEQx**		**ADC12BUSY**

The **bold** bits have the same functionality as in the ADC10. Refer to the ADC10 section for more details. As in the previous register, the following table only contains the differences from the ADC10.

Bit		Description
15–12	CSTARTADDx	Conversion start address. These bits select which ADC12MEMx is used for a single conversion or for the first conversion in a sequence. The value of CSTARTADDx is 0 to 0Fh, corresponding to ADC12MEM0 to ADC12MEM15.
9	SHP	Sample-and-hold mode select: SHP = 0 ⇒ SAMPCON signal is sourced from the sample-input signal SHP = 1 ⇒ SAMPCON signal is sourced from the sampling timer

ADC12MEMx, Conversion-Memory Register

These 16-bit registers will be loaded with the conversion results. Bits 15-12 are always 0. The conversion results are stored in the least significant 12 bits, Bit 11 being the MSB.

ADC12MCTLx, ADC12 Conversion Memory Control Registers

7	6	5	4	3	2	1	0
EOS	SREFx			INCHx			

The INCHx depends on the device. See specific datasheet.

Bit		Description		
7	EOS	Indicates the last conversion in a sequence: EOS = 0 ⇒ Not end of sequence EOS = 1 ⇒ End of sequence		
6-4	SREFx	Select voltage reference:	V_{R+}	V_{R-}
		SREF2 SREF1 SREF0 = 000 ⇒	AV_{CC}	AV_{SS}
		SREF2 SREF1 SREF0 = 001 ⇒	V_{REF+}	AV_{SS}
		SREF2 SREF1 SREF0 = 010 ⇒	Ve_{REF+}	AV_{SS}
		SREF2 SREF1 SREF0 = 011 ⇒	Ve_{REF+}	AV_{SS}
		SREF2 SREF1 SREF0 = 100 ⇒	AV_{CC}	V_{REF-}/Ve_{REF-}
		SREF2 SREF1 SREF0 = 101 ⇒	V_{REF+}	V_{REF-}/Ve_{REF-}
		SREF2 SREF1 SREF0 = 110 ⇒	Ve_{REF+}	V_{REF-}/Ve_{REF-}
		SREF2 SREF1 SREF0 = 111 ⇒	Ve_{REF+}	V_{REF-}/Ve_{REF-}
3-0	INCHx	Input channel select: INCH3 INCH2 INCH1 INCH0 = 0000 ⇒ A0 INCH3 INCH2 INCH1 INCH0 = 0001 ⇒ A1 INCH3 INCH2 INCH1 INCH0 = 0010 ⇒ A2 INCH3 INCH2 INCH1 INCH0 = 0011 ⇒ A3 INCH3 INCH2 INCH1 INCH0 = 0100 ⇒ A4 INCH3 INCH2 INCH1 INCH0 = 0101 ⇒ A5 INCH3 INCH2 INCH1 INCH0 = 0110 ⇒ A6 INCH3 INCH2 INCH1 INCH0 = 0111 ⇒ A7 INCH3 INCH2 INCH1 INCH0 = 1000 ⇒ Ve_{REF+} INCH3 INCH2 INCH1 INCH0 = 1001 ⇒ V_{REF-}/Ve_{REF-} INCH3 INCH2 INCH1 INCH0 = 1010 ⇒ Temperature sensor INCH3 INCH2 INCH1 INCH0 = 1011 ⇒ $(AV_{CC} - AV_{SS})/2$ INCH3 INCH2 INCH1 INCH0 = 1100 ⇒ A12 INCH3 INCH2 INCH1 INCH0 = 1101 ⇒ A13 INCH3 INCH2 INCH1 INCH0 = 1110 ⇒ A14 INCH3 INCH2 INCH1 INCH0 = 1111 ⇒ A15		

ADC12IE, ADC12 Interrupt Enable Register

This 16-bit register enables (ADC12IEx = 1) or disables (ADC12IEx = 0) the interrupt request for the ADC12IFGx bits.

ADC12IFG, ADC12 Interrupt Flag Register

Each bit of this 16-bit register is set when the corresponding ADC12MEMx is loaded with a conversion result and reset if the corresponding ADC12MEMx is accessed by software.

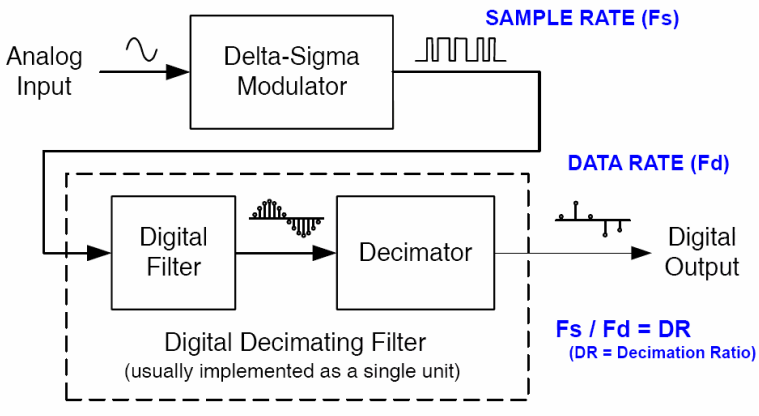
9.3.4 Sigma-Delta (SD) converter

Sigma-Delta (SD) converter determines the digital word by the steps shown in the block diagram given in *Figure 9-44*:

- ❑ Oversampling the input signal by sigma-delta modulator;
- ❑ Applying digital filtering;
- ❑ Reducing data rate by collecting modulator output bits (Decimation).

In the next section, the individual components of a SD converter will be described. To simplify this presentation, it uses a simple sine wave analogue input, and all the components are reduced to first order.

Figure 9-44. SD converter – Block diagram.

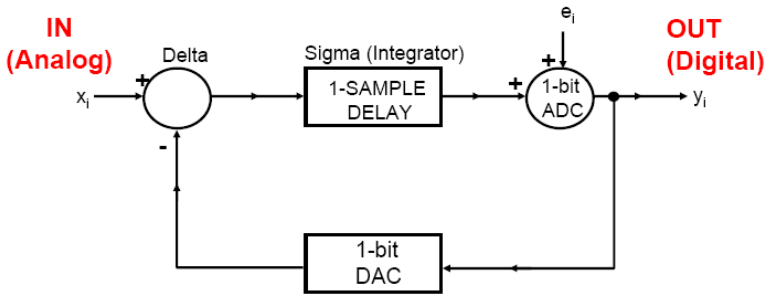


Delta modulator

Delta modulation quantizes the difference between the current analogue input signal and the average of the previous ones. In its simplest form, it is a first order modulator, (see *Figure 9-45*). The quantization can be realized as a comparator, referenced to 0 (two

levels quantized), whose output is 1 or 0 when the analogue input signal is positive or negative. The demodulator is simply an integrator (1-bit DAC in the feedback loop), whose output rises or falls with each 1 or 0 received, maintaining the average output of the integrator near to the reference level of the comparator.

Figure 9-45. SD converter – 1st order Delta modulator block diagram.



$$y_i = x_{i-1} + (e_i - e_{i-1})$$

The density of "ones" at the modulator output is proportional to the input signal:

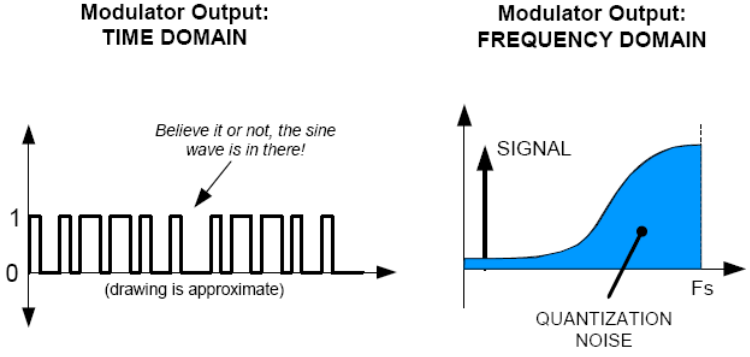
- For an increasing analogue input, the comparator generates a greater number of "ones";
- For a decreasing analogue input, the comparator generates a lesser number of "ones".

By summing the error voltage, the integrator acts as a:

- Lowpass filter to the input signal;
- Highpass filter to the quantization noise.

Due to this configuration, most of the quantization noise is pushed into higher frequencies. Oversampling has changed not only the total noise power, but its distribution has been altered (see the AC performance parameters at the beginning of the Analogue-to-Digital Conversion chapter). The output of the delta modulator is shown for both time and frequency domains in *Figure 9-46*. It includes the quantization noise that limits the dynamic range of the ADC. This noise is actually the "round-off" error that occurs when an analogue signal is quantized.

Figure 9-46. SD converter – Delta modulator output: Time and frequency domains.



Note that the MSP430 includes a 2nd order Delta-Sigma Modulator, with a block diagram as shown in *Figure 9-47*.

Figure 9-47. SD converter – 2nd order SD modulator.

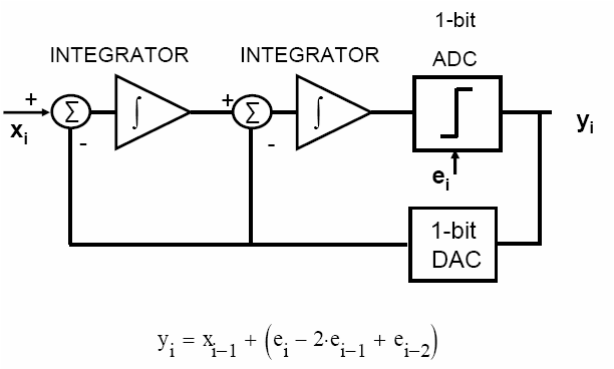
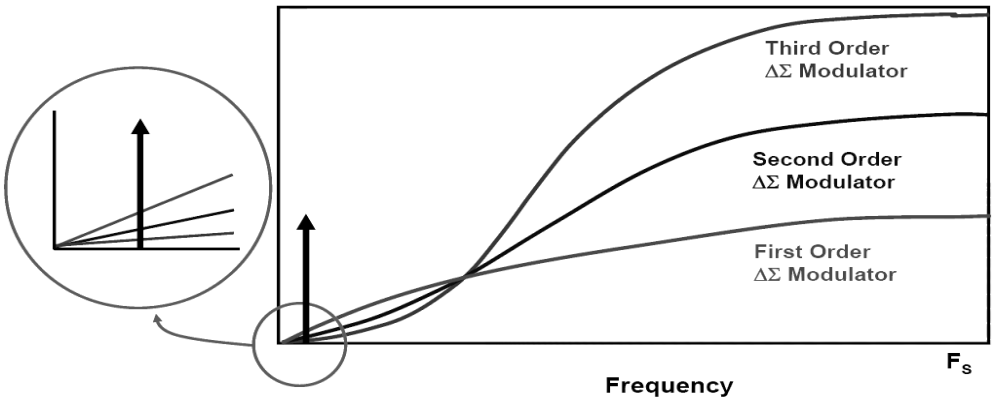


Figure 9-48 illustrates that as the OSR (Over-Sampling Ratio) increases, the noise decreases (SNR increases) and that as the order of the modulator increases, the noise decreases.

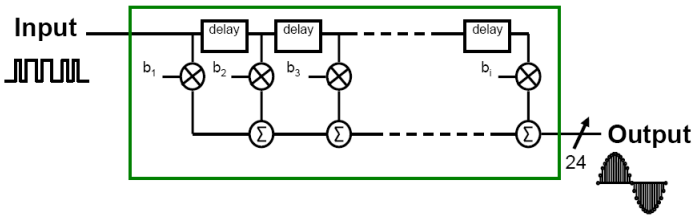
Figure 9-48. SD converter – Multi order SD modulator.



Digital filter

A digital filter, whose block diagram is shown in *Figure 9-49,* is used to attenuate signals and noise that are outside the band of interest. Applying a digital filter to the noise-shaped delta-sigma modulator, removes more noise than does simple oversampling. A 1st order modulator provides a 9dB improvement in SNR for every doubling of the sampling rate.

Figure 9-49. SD converter – Digital filter block diagram.



The digital filter averages the 1-bit data stream, improves the analogue-to-digital conversion resolution, and removes quantization noise that is outside the band of interest. It determines the signal bandwidth, settling time, and stopband rejection.

It should be remembered that with SD ADCs, the resolution is different from the data word number of bits *n* and usually it is expressed as function of the Effective Number Of Bits (ENOB). The resolution also depends on the Oversampling Ratio (OSR).

There are several types of digital filters:

- Finite Impulse Response (FIR) filter, also known as a non-recursive filter (output is dependent only on past and present values of the input);
- Sinc filter, ideally it removes all frequency components above a given bandwidth, remains the low frequencies components, and has linear phase;
- Infinite Impulse Response (IIR) filter, also known as a recursive filter (output is dependent on past and present values of both the input and the output);
- Averaging, Moving average filter.

In SD converters, a widely used filter topology that performs the lowpass function is the Sinc³ or Sinc^5 types (see *Figure 9-50*). The main advantages of these filters are their notch responses. The notch position is directly related to the output data rate, allowing the reduction of the high frequency noise as shown in *Figure 9-51*.

Figure 9-50. SD converter – Sinc^3 and Sinc^5 digital filters.

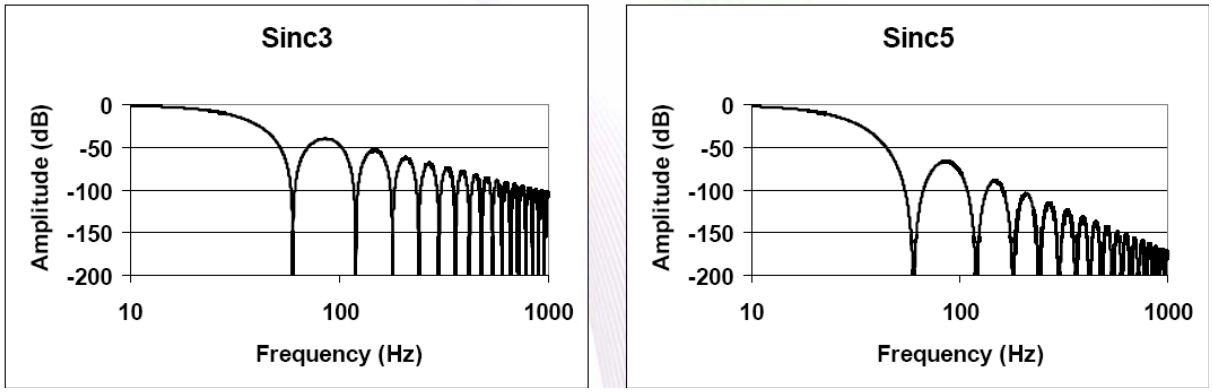
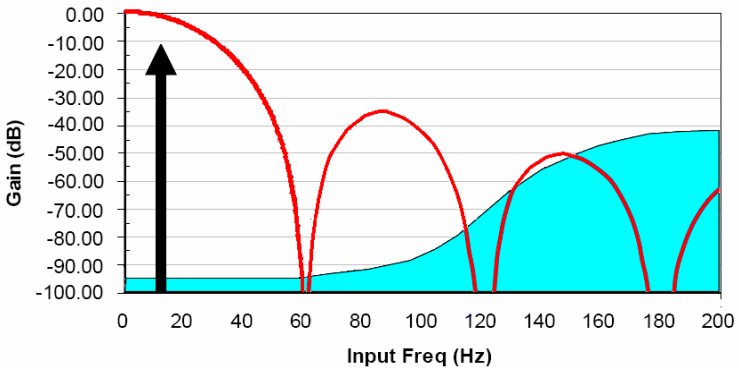


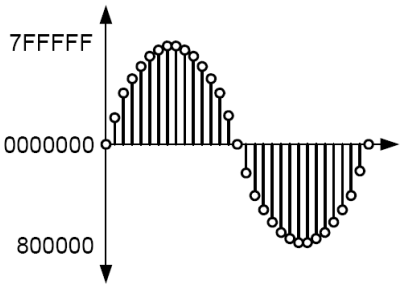
Figure 9-51. SD converter – Sinc^3 high frequency noise reduction.



Here the notch frequencies have been chosen to occur at multiples of 60Hz, to remove mains frequency noise and harmonics.

The output of the digital filter will be a data stream as shown in *Figure 9-52*.

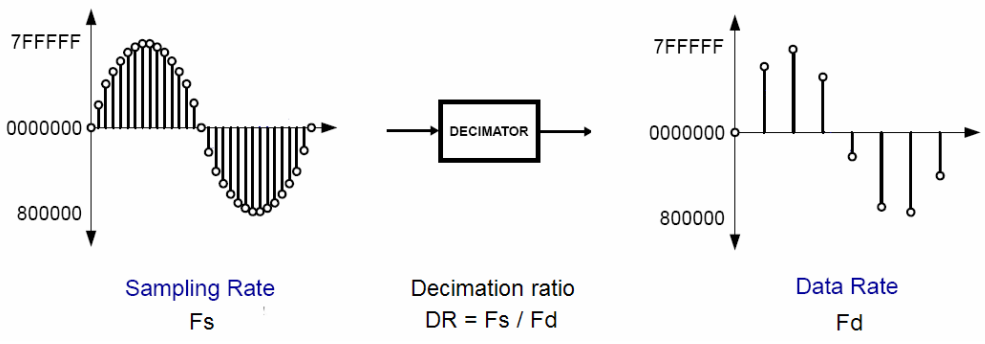
Figure 9-52. SD converter – Digital filter output: data stream.



Decimation digital filter

Decimation consists of reducing the data rate down from the oversampling rate, without losing information, thus eliminating redundant data at the output. Using the Nyquist theorem ($f_{sample} > 2 \times f_{input}$), and by the oversampling at the delta modulator, the input signal can be reliably reconstructed without distortion. The decimation digital filter will preserve certain input samples and discard others based on a data rate, which is smaller than the sampling rate, as shown in *Figure 9-53*.

Figure 9-53. SD converter – Decimation digital filter: sampling rate vs. data rate.



MSP430 SD16(A) – Sigma/Delta Analogue-to-Digital Converter

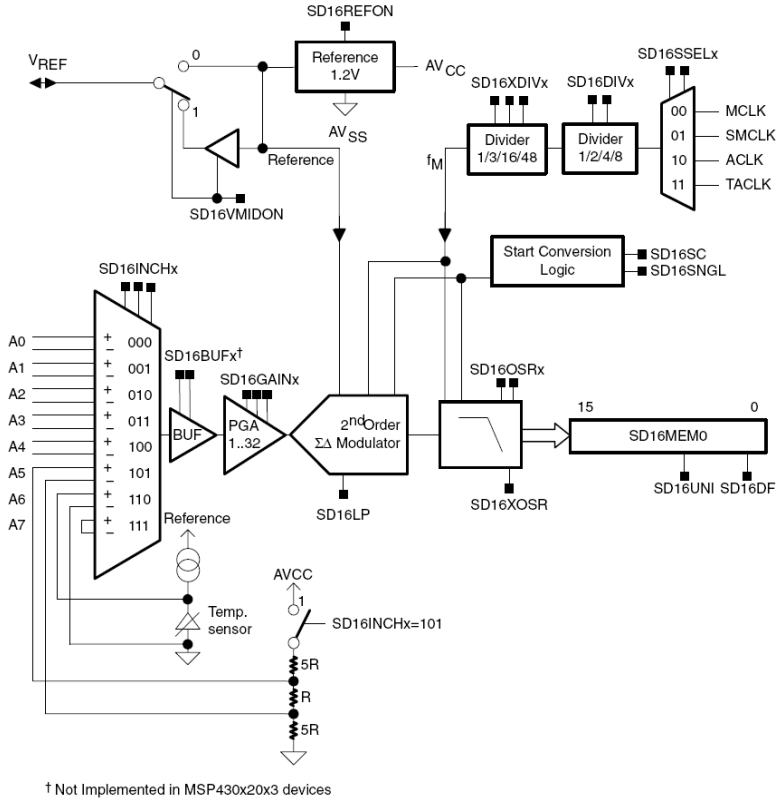
This section gives details of the Sigma/Delta converter contained in the MSP430 devices. Some devices dedicated to particular applications, such as portable medical (F42xx and FG42xx), energy metering (FE42x(A), F47xx and F471xx) and generic application (F42x and F20x3), contain up to 7 SD converters. See the TI web page for details.

The eZ430-F2013 hardware development tool includes a SD ADC. The module integrated into this device is the SD16_A that supports 16-bit analogue-to-digital conversions. The module implements a 16-bit SD core and reference generator. In addition to external analogue inputs, internal V_{CC} sense and temperature sensors are also available.

The basic features of the SD16_A module included in the eZ430-F2013 are:

- 16-bit sigma-delta architecture;
- Up to eight multiplexed differential analogue inputs per channel;
- Software selectable on-chip reference voltage generation (1.2 V);
- Software selectable internal or external reference;
- Built-in temperature sensor;
- Up to 1.1 MHz modulator input frequency;
- Selectable low-power conversion mode.

Figure 9-54. SD16_A block diagram.



❑ *SD ADC core*

The analogue-to-digital conversion is performed by a 1-bit, 2nd oversampling sigma-delta modulator.

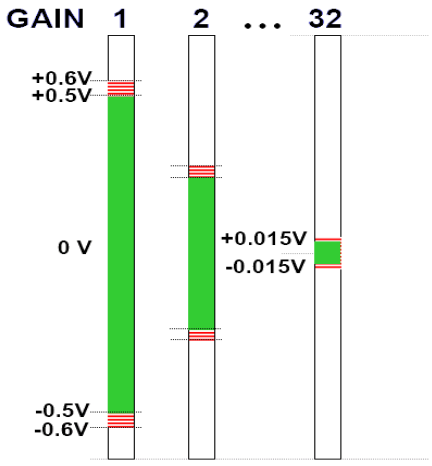
A single-bit comparator in the modulator quantizes the input signal at the modulator frequency, f_M. The resulting 1-bit data stream is averaged by the digital decimation filter (comb type filter with selectable oversampling), to obtain the conversion result. The decimation filter has ratios of up to 1024. Additional filtering can be performed in software.

❑ *Analogue input range and PGA*

The full-scale input voltage range for each analogue input pair is dependent on the gain setting of the programmable gain amplifier (PGA= 1, 2, 4, 8, 16 and 32) of each channel (applies to all inputs and modes). The maximum full-scale range is $\pm V_{FS}$ where V_{FS} is defined by the following equation (see *Figure 9-55*):

$$V_{FS} = \frac{V_{ref}/2}{GAIN_{PGA}}$$

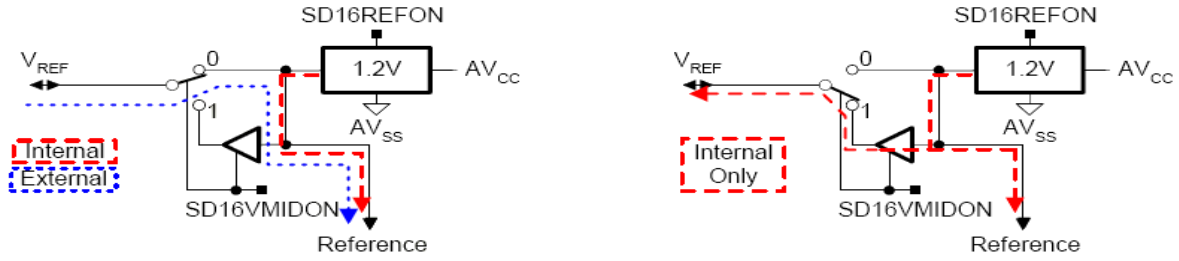
Figure 9-55. SD16_A analogue input range.



❑ *Voltage reference generator*

The SD16_A module can use a 1.2 V internal voltage reference, enabled by the SD16REFON bit (see *Figure 9-56*). To reduce noise, it is recommended that an external 100 nF capacitor is connected between V_{REF} and AV_{SS}.

Figure 9-56. SD16_A Voltage reference generator.



Options for reference voltage connections:

- External reference:
  - SD16REFON = 0, SD16VMIDON = 0;
- Internal reference:
  - SD16REFON = 1, SD16VMIDON = 0;
- Off-chip internal reference with buffered output (can drive 1mA):
  - SD16REFON = 1, SD16VMIDON = 1;
  - Requires a 470 nF capacitor connected between V_{REF} and AV_{SS}.
- For internal temperature sensor (A6): use the internal reference.

❑ *Auto power-down*

For low power applications, the SD16_A is automatically disabled when it is not actively converting and automatically re-enabled when a conversion is started. The reference can be disabled by software (resetting SD16REFON bit).

❑ *Analogue input pair selection*

The SD16_A can convert up to 8 differential input pairs multiplexed into the PGA.

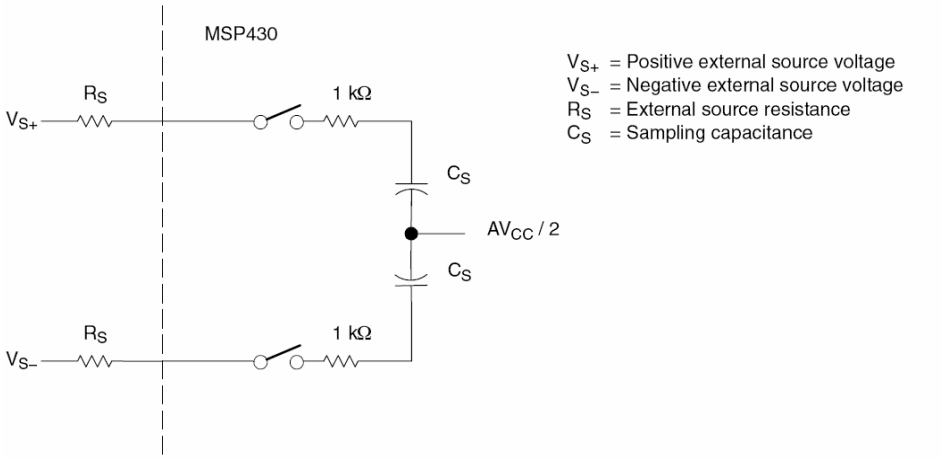
The available analogue input pairs are:

- A0-A4: Externally on the device;
- A5: Resistive divider to measure the supply voltage;
- A6: Internal temperature sensor;
- A7: Offset shunt (used for calibration of SD16_A input PGA offset measurement).

❑ *Analogue input characteristics and setup*

The analogue input equivalent circuit for the eZ430-F2013 is shown in *Figure 9-57*.

Figure 9-57. Analogue input equivalent circuit for the eZ430-F2013.



The maximum sampling frequency, f_S, is calculated from the minimum settling time, $t_{Settling}$, of the sampling circuit:

$$t_{Settling} \geq (R_S + 1k\Omega) \times C_S \times \ln\left(\frac{GAIN \times 2^{17} \times V_{Ax}}{V_{REF}}\right)$$

$$f_S = \frac{1}{2 \times t_{Settling}}$$

$$V_{Ax} = \max\left\{ \left| \frac{AV_{CC}}{2} - V_{S+} \right| , \left| \frac{AV_{CC}}{2} - V_{S-} \right| \right\}$$

With V_{S+} and V_{S-} referenced to AV_{SS}.

The sampling capacitance, C_S, varies with the gain setting as shown in *Table 9-7*.

Table 9-7. Sampling capacitance vs. gain.

PGA gain	C_S
1	1.25 pF
2	2.5 pF
4	2.5 pF
8	5 pF
16	10 pF
32	10 pF

The analogue input is configured using the SD16INCTL0 and the SD16AE registers.

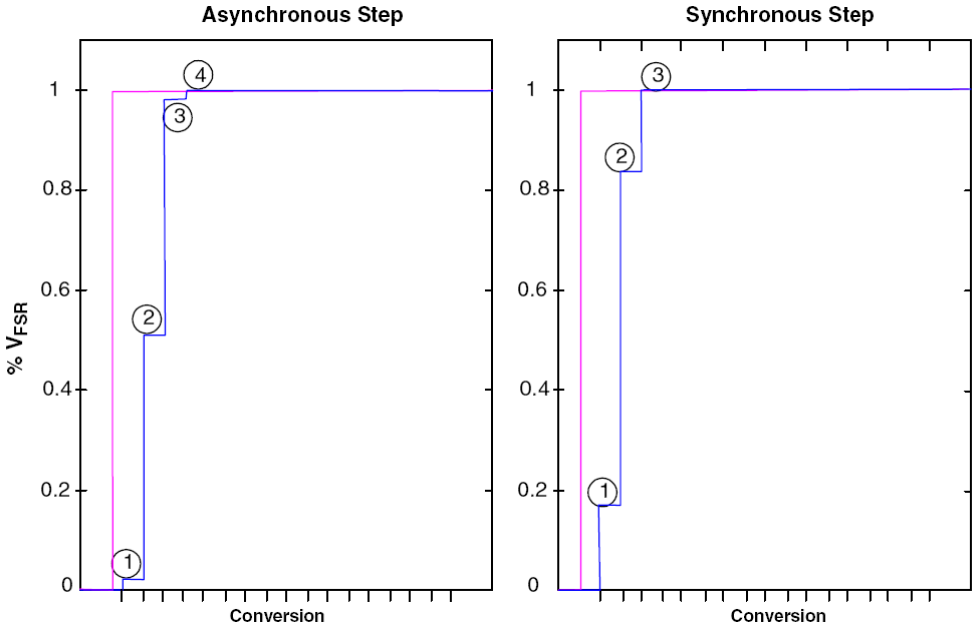
Their bits are dedicated to configure:

- SD16INCHx: One of eight differential input pairs of the analogue multiplexer;
- SD16GAINx: Gain of 1, 2, 4, 8, 16 and 32;
- SD16AEx: Enable or disable the analogue input pin;

Note: During conversion, any modification to the SD16INCHx and SD16GAINx bits will become effective at the next decimation step of the digital filter. Since the digital filter is a $Sinc^3$ comb type, it will require three data-word periods to settle, as shown in *Figure 9-58,* for asynchronous and synchronous conversion steps.

If the SD16INTDLY bits are set to 00h, conversion interrupt requests do not begin until the 4^{th} conversion after a start condition.

Figure 9-58. Digital filter step response and conversion points.



- ***Digital filter***

The digital filter processes the 1-bit data stream from the modulator using a $Sinc^3$ comb filter.

In this case the:

- Oversampling rate is given by: OSR = f_M / f_S
- First filter notch is at: f_S = f_M / OSR

To adjust the notch frequency, the following registers must be configured:

- SD16SSELx and SD16DIVx: to change the modulator's frequency, f_M;
- SD16OSRx and SD16XOSR bits: to change the oversampling rate, OSR.

The number of bits output by the digital filter is dependent on the oversampling ratio, decimation ratio and number format, ranging from 15 to 30 bits. See the corresponding MSP430 User's Guide for additional details.

- ***Conversion memory register (SD16MEM0)***

Conversion results are moved to the SD16MEM0 register with each decimation step of the digital filter.

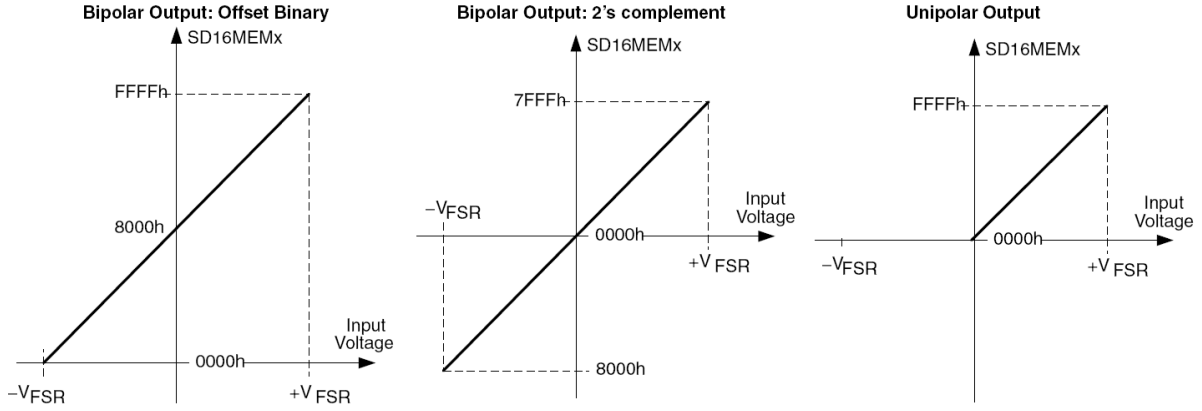
The S16IFG bit (SD16 flag) is modified by:

- New data is written to SD16MEM0: SD16IFG = 1;
- SD16MEM0 read by CPU or cleared by software: SD16IFG = 0.

- ***Output data format***

Selecting the SD16DF and SD16UNI bits, the output data format can be configured in two's complement, offset binary or unipolar mode, as shown in *Figure 9-59.*

Figure 9-59. Input voltage vs. Digital output.



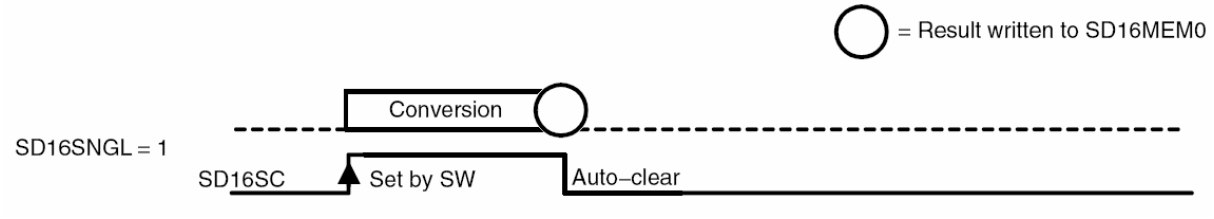
SD16UNI = 0	SD16UNI = 0	SD16UNI = 1
SD16DF = 0	SD16DF = 1	SD16DF = 0

- ***Conversion modes***

The SD16_A module can be configured for two modes of operation:

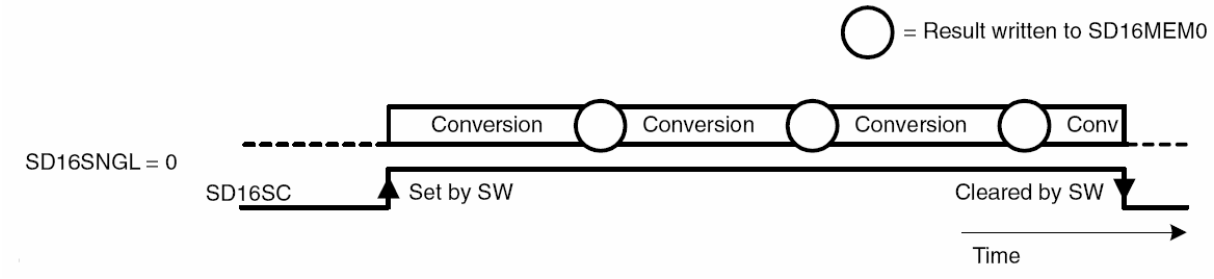
- Single conversion (see *Figure 9-60*):
  - The channel is converted once (SD16SNGL = 1);
  - After conversion is completed: SD16SC = 0;
  - Clearing SD16SC before the conversion is completed:
    - Immediately stops conversion of the channel;
    - Powers down the channel;
    - Turns off the corresponding digital filter;
    - The value in SD16MEM0 can change.

Figure 9-60. Single conversion mode.



- Continuous conversion (see *Figure 9-61*):
  - The channel is converted continuously (SD16SNGL = 0);
  - Starts when SD16SC = 1;
  - Stops when SD16SC = 0 (cleared by software);
  - Clearing SD16SC before the conversion is completed has the same effect as for single conversion.

Figure 9-61. Continuous conversion mode.



- ***Integrated temperature sensor***

Configured when:

- Analogue input pair SD16INCHx = 110;

- SD16REFON = 1;
- SD16VMIDON = 1 (if is used an external reference for other analogue input pair).

The typical temperature sensor transfer function is given by:

$V_{Sensor,typ} = T_{CSensor}(273 + T[ºC]) + V_{Offset,\ sensor}$ [mV]

- ***SD16_A interrupts***

Two interrupt sources for its ADC channel:

- SD16 Interrupt Flag (SD16IFG):
  - SD16IFG = 1: SD16MEM0 memory register is written with a conversion result;
  - An interrupt request requires:
    - Corresponding SD16IE = 1;
    - GIE = 1.
- SD16 Overflow Interrupt Flag (SD16OVIFG):
  - SD16OVIFG = 1: conversion result is written to SD16MEM0 location before the previous conversion result was read.

- ***Interrupt vector generator (SD16IV)***

The interrupt vector register SD16IV is used to determine which enabled SD16_A interrupt source requested an interrupt. The following factors are taken into consideration by the SD16_A interrupt procedure:

- The highest priority SD_16 interrupt enabled generates a number in the SD16IV register, which can be evaluated or added to the program counter to automatically enter the appropriate software routine. Disabled SD_16 interrupts do not affect the SD16IV value.
- Any access, read or write, of the SD16IV register has no effect on the SD16OVIFG or SD16IFG flags. The SD16IFG flags are reset by reading the SD16MEM0 register or by clearing the flags in software. SD16OVIFG bits can only be reset by software.
- If another interrupt is pending after servicing of an interrupt, another interrupt is generated depending on the priority.

- ***Interrupt delay operation***

In order to allow the digital filter to settle prior to generating an interrupt request, the SD16INTDLYx bits set the interrupt request delay (up to four conversion cycles). The SD16OVIFG bits for the selected number of delay cycles are also reset.

The delay is applied each time:

- SD16SC = 1;
- SD16GAINx or SD16INCHx bits for the channel are modified.

- ***SD16 registers***

SD16CTL, SD16_A Control Register

15	14	13	12	11	10	9	8
Reserved				SD16XDIVx			SD16LP

7	6	5	4	3	2	1	0
SD16DIVx		SD16SSELx		SD16VMIDON	SD16REFON	SD16OVIE	Reserved

Bit		Description
11-9	SD16XDIVx	SD16_A clock divider: SD16XDIV2 SD16XDIV1 SD16XDIV0 = 000 ⇒ /1 SD16XDIV2 SD16XDIV1 SD16XDIV0 = 001 ⇒ /3 SD16XDIV2 SD16XDIV1 SD16XDIV0 = 010 ⇒ /16 SD16XDIV2 SD16XDIV1 SD16XDIV0 = 011 ⇒ /48 SD16XDIV2 SD16XDIV1 SD16XDIV0 = 1xx ⇒ Reserved
8	SD16LP	Low power mode enable when SD16LP = 1
7-6	SD16DIVx	SD16_A clock divider: SD16DIV1 SD16DIV0 = 00 ⇒ /1 SD16DIV1 SD16DIV0 = 01 ⇒ /2 SD16DIV1 SD16DIV0 = 10 ⇒ /4 SD16DIV1 SD16DIV0 = 11 ⇒ /8
5-4	SD16SSELx	SD16_A clock source: SD16SSEL1 SD16SSEL0 = 00 ⇒ MCLK SD16SSEL1 SD16SSEL0 = 01 ⇒ SMCLK SD16SSEL1 SD16SSEL0 = 10 ⇒ ACLK SD16SSEL1 SD16SSEL0 = 11 ⇒ External TACLK
3	SD16VMIDON	V_{MID} buffer on when SD16VMIDON = 1
2	SD16REFON	Reference generator on when SD16REFON = 1
1	SD16OVIE	SD16_A overflow interrupt enable when SD16OVIE = 1 (GIE bit must also be set)

SD16CCTL0, SD16_A Control Register 0

15	14	13	12	11	10	9	8
Reserved			SD16UNI	SD16XOSR	SD16SNGL	SD16OSRx	

7	6	5	4	3	2	1	0
SD16LSBTOG	SD16LSBACC	SD16OVIFG	SD16DF	SD16IE	SD16IFG	SD16SC	Reserved

Bit		Description
12	SD16UNI	Unipolar mode: SD16UNI = 0 ⇒ Bipolar mode SD16UNI = 1 ⇒ Unipolar mode
11	SD16XOSR	Extended oversampling ratio. This bit, along with the SD16OSRx bits, selects the oversampling ratio.
10	SD16SNGL	Conversion mode: SD16SNGL = 0 ⇒ Continuous conversion mode SD16SNGL = 1 ⇒ Single conversion mode
9-8	SD16OSRx	Oversampling ratio: SD16XOSR = 0 SD16XOSR = 1 SD16OSR1 SD16OSR0 = 00 ⇒ 256 512 SD16OSR1 SD16OSR0 = 01 ⇒ 128 1024 SD16OSR1 SD16OSR0 = 10 ⇒ 64 Reserved SD16OSR1 SD16OSR0 = 11 ⇒ 32 Reserved
7	SD16LSBTOG	When SD16LSBTOG = 1 causes SD16LSBACC to toggle each time the SD16MEM0 register is read.
6	SD16LSBACC	LSB access: SD16LSBACC = 0 ⇒ SD16MEMx contains the most significant 16 bits of the conversion result SD16LSBACC = 1 ⇒ SD16MEMx contains the least significant 16-bits of the conversion result
5	SD16OVIFG	SD16_A overflow interrupt flag SD16OVIFG = 1 indicates an overflow interrupt pending
4	SD16DF	SD16_A data format: SD16DF = 0 ⇒ Offset binary SD16DF =1 ⇒ Two's complement
3	SD16IE	SD16IE = 1 enables SD16_A interrupt
2	SD16IFG	SD16_A interrupt flag: SD16IFG = 0 ⇒ corresponding SD16MEMx register is read and no interrupt pending SD16IFG = 1 ⇒ when it is an interrupt pending (new conversion results available)
1	SD16SC	SD16SC = 1 ⇒ start conversion with the SD16_A

SD16INCTL0, SD16_A Input Control Register

7	6	5	4	3	2	1	0
SD16INTDLYx		SD16GAINx			SD16INCHx		

Bit		Description
7-6	SD16INTDLYx	Interrupt delay generation after conversion start: SD16INTDLY1 SD16INTDLY0 = 00 ⇒ 4th sample causes interrupt SD16INTDLY1 SD16INTDLY0 = 01 ⇒ 3rd sample causes interrupt SD16INTDLY1 SD16INTDLY0 = 10 ⇒ 2nd sample causes interrupt SD16INTDLY1 SD16INTDLY0 = 11 ⇒ 1st sample causes interrupt
5-3	SD16GAINx	SD16_A PGA gain: SD16GAIN2 SD16GAIN1 SD16GAIN0 = 000 ⇒ x1 SD16GAIN2 SD16GAIN1 SD16GAIN0 = 001 ⇒ x2 SD16GAIN2 SD16GAIN1 SD16GAIN0 = 010 ⇒ x4 SD16GAIN2 SD16GAIN1 SD16GAIN0 = 011 ⇒ x8 SD16GAIN2 SD16GAIN1 SD16GAIN0 = 100 ⇒ x16 SD16GAIN2 SD16GAIN1 SD16GAIN0 = 101 ⇒ x32 SD16GAIN2 SD16GAIN1 SD16GAIN0 = 110 ⇒ Reserved SD16GAIN2 SD16GAIN1 SD16GAIN0 = 111 ⇒ Reserved
2-0	SD16INCHx	SD16_A channel differential pair input: SD16INCH2 SD16INCH1 SD16INCH0 = 000 ⇒ A0 SD16INCH2 SD16INCH1 SD16INCH0 = 001 ⇒ A1 SD16INCH2 SD16INCH1 SD16INCH0 = 010 ⇒ A2 SD16INCH2 SD16INCH1 SD16INCH0 = 011 ⇒ A3 SD16INCH2 SD16INCH1 SD16INCH0 = 100 ⇒A4 SD16INCH2 SD16INCH1 SD16INCH0 = 101 ⇒ A5: (AV_{CC}–AV_{SS})/11 SD16INCH2 SD16INCH1 SD16INCH0 = 110 ⇒ A6: Temp. sensor SD16INCH2 SD16INCH1 SD16INCH0 = 111 ⇒ A7: Offset shunt

SD16MEM0, SD16_A Conversion Memory Register

The 16 bits of the SD16MEMx register store the conversion results. Depending on the SD16LSBACC bit state, they holds the upper or lower 16 bits of the digital filter output.

SD16AE, SD16_A Analogue Input Enable Register

The 8 bits of SD16_A analogue enable register must be set (SD16AE = 1) to enable the corresponding external analogue input (MSB: A7 to LSB: A0).

SD16IV, SD16_A Interrupt Vector Register

The 16 bits of the SD16IV interrupt vector register, depending on the content of bits 1 to 4, indicate the interrupt source. According to their priority:

- SD16IV = 002h ⇒ SD16MEMx overflow;
- SD16IV = 004h ⇒ SD16_A Interrupt.

For SD16IV = 000h there is no interrupt pending.

From SD16IV = 006h to =010h (lowest) the interrupt source is reserved.

9.3.5 Comparator-Based Slope ADC

Analogue-to-digital conversion by integration is based on counting clock pulses. An ADC that uses an integration method is known as a Slope ADC. This technique can be implemented with a comparator rather than a standalone ADC module or device. The integration method relies on the charging/discharging of a capacitor and counting the number of clock cycles to do so. Longer discharge times indicate larger voltages. The voltage is derived from the discharge time using the standard equation for capacitor discharge.

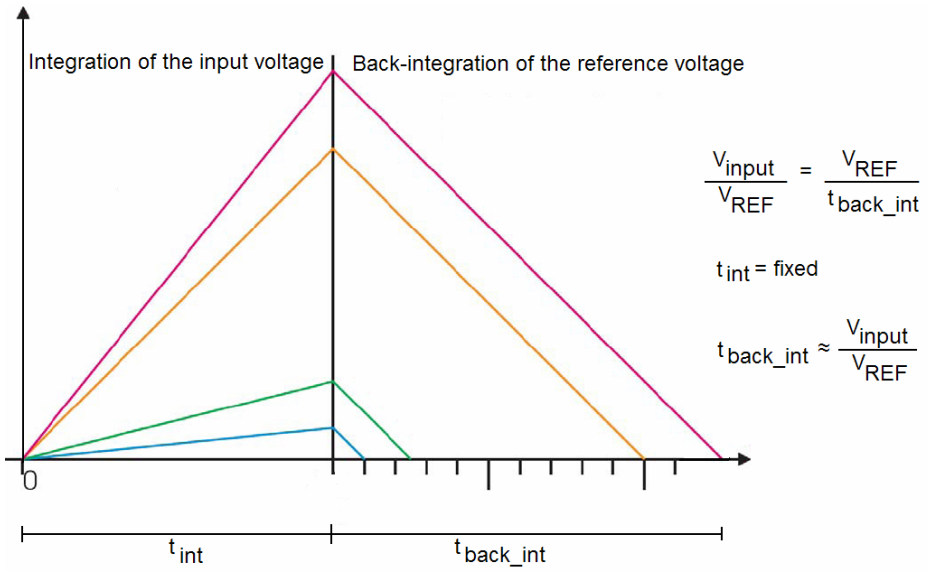
Additionally, this method can be used to measure resistance (potentiometers, transducers, etc.), because the resistance and time relationship is linear.

Single and dual slope ADC

The simplest form of analogue-to-digital conversion by integration uses single slope architecture. The method consists integration of the unknown input voltage and value comparison with a known reference value. The time it takes till the two voltages are equal is proportional to the unknown voltage. However, the accuracy of this method is dependent on the tolerance of the passive elements (resistors and capacitors), which vary with the environment, resulting in low measurement repeatability.

The dual slope integration method overcomes this difficulty, because it consists of integration of the unknown input voltage, V_{input}, for a fixed amount of time, t_{int}, and the back-integration of the known reference voltage, V_{REF}, for a variable amount of time, t_{back_int} (see Figure *9-62*).

Figure 9-62. Dual slope integration method.



This method requires a:

- ❑ Switch;
- ❑ Clock;
- ❑ Timer;
- ❑ Comparator.

The resolution depends on the clock frequency and ramp duration.

Some MSP430 devices have no true ADC, but they do have analogue comparator module (comparator_A), that can be used to make a low power slope ADC.

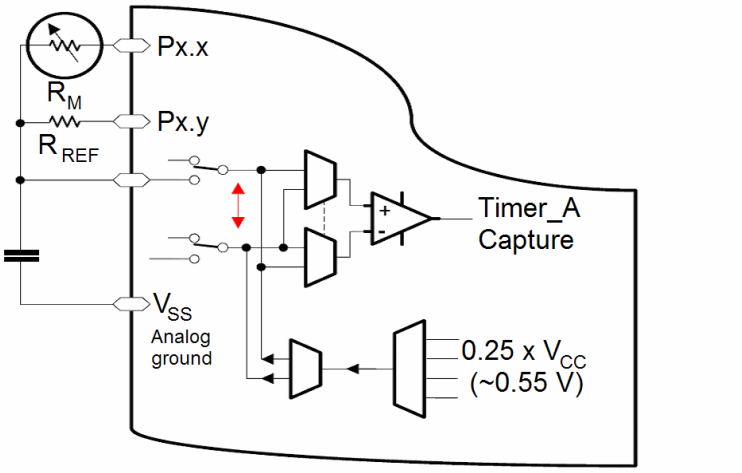
The Comparator_A is included in the MSP430FG4618 device found on the Experimenter's board. The following sub-chapter is dedicated to Comparator_A module. This section will only contain information on how to use this module as a slope ADC.

Resistive sensors measurements

The Comparator_A can be used to measure resistive elements using single slope analogue-to-digital conversion.

A thermistor is a type of resistor with resistance varying according to its temperature. So, the temperature can be converted into digital data by comparing the thermistor, R_M, capacitor discharge time to that of a reference resistor, R_{REF}. *Figure 9-63* shows the schematic diagram of the temperature measurement system.

Figure 9-63. Temperature measurement system.



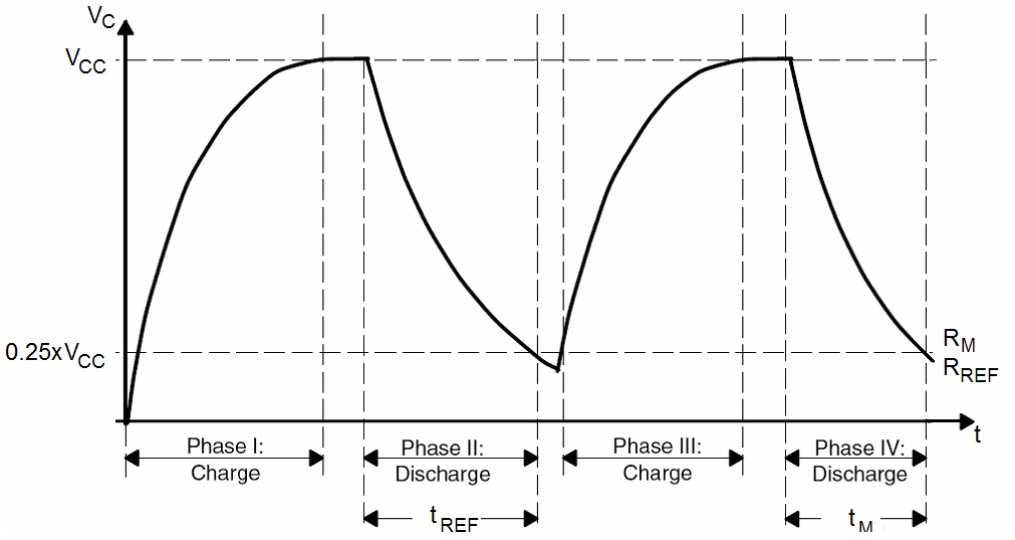
The MSP430 resources used to calculate the temperature sensed by R_M are:

- ❑ Two digital I/O pins (Px.x and Px.y) to charge and discharge the capacitor;
- ❑ I/O set to output high (V_{CC}) to charge capacitor, reset to discharge;
- ❑ I/O switched to high-impedance input with CAPDx set when not in use;
- ❑ One output charges and discharges the capacitor via R_{REF};
- ❑ One output discharges capacitor via R_M;
- ❑ The + terminal is connected to the positive terminal of the capacitor;
- ❑ The – terminal is connected to a reference level (for example V_{CAREF} = 0.25 x V_{CC});
- ❑ The output filter should be used to minimize switching noise;
- ❑ CAOUT used to gate Timer_A CCI1B, capturing capacitor discharge time.

In the equation that describes the capacitor discharge time, the thermistor measurement is based on a ratiometric conversion principle (see *Figure 9-64*):

$$t_x = -R_x \times C \times \ln\left(\frac{V_{REF}}{V_{CC}}\right)$$

Figure 9-64. Charge/Discharge Timing for Temperature Measurement System.



The ratio of two capacitor discharge times for the two resistive elements is given by:

$$\frac{t_M}{t_{REF}} = \frac{-R_M \times C \times \ln\left(\frac{V_{REF}}{V_{CC}}\right)}{-R_{REF} \times C \times \ln\left(\frac{V_{REF}}{V_{CC}}\right)} \quad \Leftrightarrow \quad \frac{t_M}{t_{REF}} = \frac{-R_M}{-R_{REF}}$$

$$R_M = R_{REF} \times \frac{t_M}{t_{REF}}$$

Slope resistance measurement factors:

- Measurement as accurate as R_{REF};
- V_{CC} independent;
- Resolution based on number of maximum counts possible;
- Precharge of *C* impacts accuracy (although there are methods to avoid errors by precharge);
- Slope measurement time is a function of *RC*;

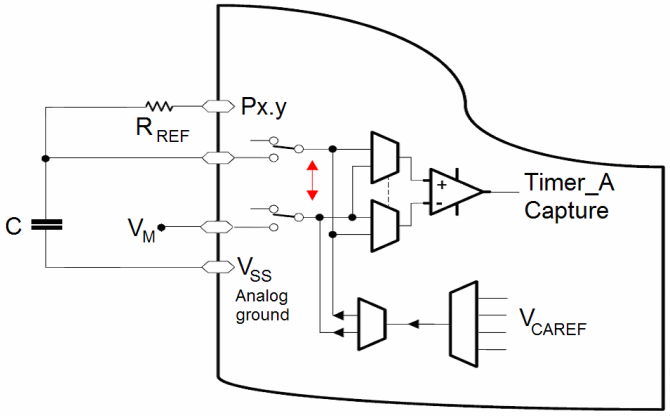
Voltage measurements

A similar application for Comparator_A module is related to voltage measurement using single slope analogue-to-digital conversion.

The voltage measurement (see *Figure 9-65*) relies on the charge or discharge of *C*. The steps for the voltage measurement are:

- Capacitor charge: $V_{SS} < V_M < V_{CAREF}$;
- Capacitor discharge: $V_{CAREF} < V_M < V_{SS}$;
- Time capture to crossing using Timer_A (TACCR1);
  - 1st: Compare to V_{CAREF};
  - 2nd: Compare to V_M.

Figure 9-65. Voltage measurement system.

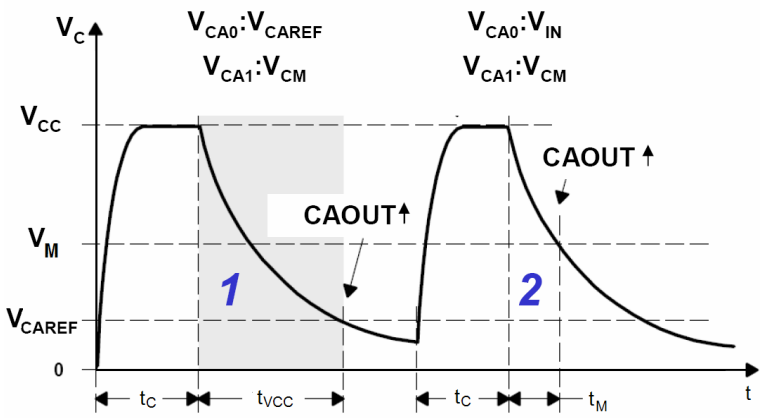


Voltage conversion and timing (see *Figure 9-66*) depends on:

- 1 Measurement:
  - $V_M = V_{REF} / e^{(-t/RC)}$;
  - V_{REF} must be stable;
  - RC tolerances influence measurements.

- 2 Measurements:
  - $V(t) = V_{CC} \times e^{(-t/RC)}$;
  - $V_M = V_{CC} \times e^{(t_M / t_{VCC}) \times \ln(0.25)}$;
  - Same approach for discharge method.

Figure 9-66. Voltage conversion and timing.



Slope voltage measurement considerations:

- The V_{CAREF} selection should maximize V_M range;
- Accuracy of result depends on V_{CC};
- Capacitor charge selection for minimum error time (7 time constant = 0.1% Error from V_{CC}).

To summarise, the comparator can be used as low-cost slope analogue-to-digital converter, only requiring a few external components. The examples cover resistance or voltage measurement and conversion into digital data, however there are other methods for measuring these analogue signals. The integration method can also be applied for current or capacitance measurements.

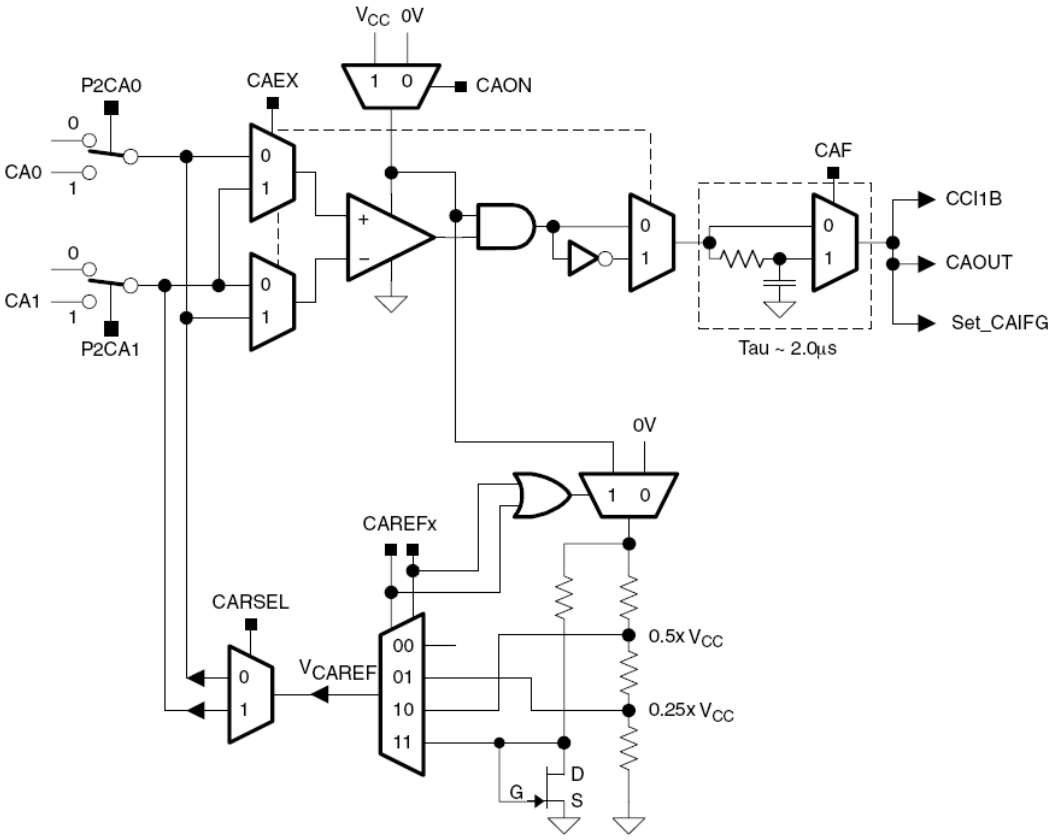
9.4 Comparator_A

As shown in the previous section, the comparator_A module is primarily designed to support precision slope analogue-to-digital conversions, battery-voltage supervision, and monitoring of external analogue signals. It is included in the FG4618 device fitted to the Experimenter's board.

Features of Comparator_A include:

- Inverting and non-inverting terminal input multiplexer;
- Software selectable low-pass filter (RC) for the comparator output;
- Outputs:
  - Timer_A capture input;
  - Interrupt (one interrupt vector with enable);
  - External.
- Software control of the port input buffer;
- Selectable references internally (reference voltage generator: V_{CAREF} = 0.25 V_{CC}; 0.5 V_{CC}; ~0.55 V –diode-) and externally;
- Comparator and reference generator can be powered down.

Figure 9-67.Comparator_A block diagram.



9.4.1 Comparator

The comparator compares the analogue voltages at the + and – input terminals, respectively CA0 and CA1 for a non-inverting topology.

The comparator output:

- CAOUT = 1:
  - CA0 > CA1.
- CAOUT = 0:
  - CA0 < CA1;
  - CAON = 0 (comparator switched off).

9.4.2 Input analogue switches

The analogue input switches individually connect or disconnect the two comparator input terminals to associated port pins, using the P2CAx bits.

9.4.3 Output filter

The output of the comparator can be used with or without internal low pass filtering, by enabling or disabling the CAF bit. Its use can reduce errors (in accuracy and resolution) associated with comparator oscillation.

9.4.4 Voltage reference generator

The CAREFx bits control the output of the voltage generator. The CARSEL bit selects the comparator terminal to which V_{CAREF} is applied.

The voltage reference generator can generate:

- V_{CAREF} = 0.25 V_{CC};
- 0.5 V_{CC};
- ~0.55 V (fixed transistor threshold voltage).

9.4.5 Comparator_A interrupts

One interrupt flag, CAIFG, and one interrupt vector are associated with the Comparator_A.

Condition for an interrupt:

- CAIFG = 1:

- Rising or falling edge of the comparator output (CAIES bits selection);
- Interrupt request when CAIE = 1 and GIE = 1.

❑ CAIFG = 0;

- Interrupt request serviced;
- With software.

9.4.6 Comparator_A registers

CACTL1, Comparator_A Control Register 1

7	6	5	4	3	2	1	0
CAEX	CARSEL	CAREFx		CAON	CAIES	CAIE	CAIFG

Bit		Description
7	CAEX	Comparator_A exchange: CAEX = 1 ⇒ Exchanges the comparator inputs and inverts the comparator output.
6	CARSEL	Comparator_A reference: CAEX = 0: CAEX = 1: CARSEL = 0 V_{CAREF} in + terminal V_{CAREF} in - terminal CARSEL = 1 V_{CAREF} in - terminal V_{CAREF} in + terminal
5-4	CAREFx	Comparator_A voltage reference: CAREF1 CAREF0 = 00 ⇒ External (Internal reference off) CAREF1 CAREF0 = 01 ⇒ V_{CAREF} = 0.25 V_{CC} CAREF1 CAREF0 = 10 ⇒ V_{CAREF} = 0.5 V_{CC} CAREF1 CAREF0 = 11 ⇒ V_{CAREF} = 0.55 V (Diode reference)
3	CAON	Comparator_A on when CAON = 1
2	CAIES	Comparator_A interrupt edge: CAIES = 0 ⇒ Rising edge CAIES = 1 ⇒ Falling edge
1	CAIE	Comparator_A interrupt enable when CAIE = 1
0	CAIFG	The Comparator_A interrupt flag, CAIFG = 1 when an interrupt is pending

CACTL2, Comparator_A Control Register 2

7	6	5	4	3	2	1	0
Unused				P2CA1	P2CA0	CAF	CAOUT

Bit		Description
3	P2CA1	The pin is connected to CA1 when P2CA1 = 1
2	P2CA0	The pin is connected to CA0 when P2CA0 = 1
1	CAF	Comparator_A output filter is enable when CAF = 1
0	CAOUT	Comparator_A output.

CAPD, Comparator_A Port Disable Register

The 8-bit Comparator_A port disable register allows individual enable or disable of each P1.x pin buffer, when the corresponding bit in CAPDx = 0 or = 1 respectively.

9.5 Laboratory 5: Signal Acquisition

This laboratory gives examples of the uses of the ADC types available in the hardware development kits. A different laboratory is developed for each kit, taking into account that both the ADC10 and the SD16_A laboratories implement a temperature data logger. The ADC12 laboratory also uses operational amplifiers to perform the analogue signal conditioning.

9.5.1 Lab5A: SAR ADC10 conversion

Project files

- C source file: **Chapter 9 > Lab5 > Lab5A_student.c**
- Solution file: **Chapter 9 > Lab5 > Lab5A_solution.c**

Overview

This laboratory implements a temperature data logger using the hardware kit's integrated temperature sensor. The device is configured to perform an acquisition each minute for one hour. Each temperature's (ºC) value is transferred to flash info memory segment B and C. When the microcontroller is not performing any task, it enters into low power mode.

A. Resources

The ADC10 module uses V_{REF+} = 1.5 V as the reference voltage.

It is necessary to configure the ADC10 to use the integrated temperature sensor (A10) as an input. Timer_A generates an interrupt once every second that starts conversion in the ADC10. At the end of a conversion, an interrupt is requested by the converter and the temperature value is written to flash memory.

The voltage value is converted into temperature following the equation provided in ADC10 sub-section. After transferring the value to the flash memory, the system returns to low power mode LPM3.

The resources used by the application are:

- ADC10;
- Timer_A;
- Ports I/O;
- Interrupts;
- Low power mode.

B. Software application organization

The application starts by stopping the Watchdog Timer.

The system checks for calibration constants on info memory segment A. The CPU execution will be trapped if it does not find this information.

Digitally controller oscillator (DCO) is set to 1 MHz, providing clock source for MCLK and SMCLK, while the Basic Clock System+ is configured to set ACLK to 1.5 kHz.

Controller's flash timing is obtained from MCLK divided by three to comply with the device specifications.

Port P1.0 is configured as an output and will blink the once LED every second.

The ADC10 is configured to use the input channel corresponding to the on-chip temperature sensor (channel A10). The configuration includes the activation of the internal reference voltage V_{REF+} = 1.5 V and the selection of ADC10OSC as clock signal. The converter is configured to perform a single-channel single-conversion. At the end of conversion, an interrupt is requested.

The Timer_A is configured to generate an interrupt once every second. ACLK/8 is selected as the clock signal using the VLOCLK as clock source and will count until the TACCR0 value is reached (in up mode). The system then enters into low power mode and waits for an interrupt.

Flash memory pointers and interrupt counters are initialized. The Timer_A ISR increments the variable counter and when this variable reaches the value 60 (1 minute), the software start of conversion is requested. At the end of this ISR, the system returns to low power mode.

When the ADC10 ends the conversion, an interrupt is requested. While variable ***min*** is lower than 60, the temperature is written to flash memory. The memory pointer is increased by two (word). When ***min*** = 60, the system stops operation.

C. System configuration

❑ DCO configuration

Adjust the DCO frequency to 1 MHz by software using the calibrated DCOCTL and BCSCTL1 register settings stored in information memory segment A.

```
if (CALBC1_1MHZ == _________ || CALDCO_1MHZ == _________)
  {
    while(1);           // If calibration constants erased
                        // do not load, trap CPU!!
  }

DCOCTL = ____________________;
```

❑ Basic Clock module+ configuration

Set MCLK and SMCLK to 1 MHz. Use the internal very low power VLOCLK source clock to ACLK/8 clock signal as low frequency oscillator (12 kHz):

```
BCSCTL1 = ____________________;
BCSCTL3 = ____________________;
```

❑ ADC10 configuration

The ADC10's input channel is the integrated temperature sensor (A10) and it uses the signal V_{REF+} (1.5 V) as reference voltage. The ADC10 clock source is ADC10OSC, the clock signal being ADC10CLK/4. Configure the ADC10 sample-and-hold time: 64xADC10CLKs, to perform a single-channel single-conversion and enable its interrupts. What are the values to write to the configuration registers?

```
ADC10CTL1 = ___________________;
ADC10CTL0 = ___________________;
```

❑ Timer_A configuration

Configure Timer_A register to enable an interrupt once every second. Use the ACLK clock signal as the clock source. This timer is configured in up counter mode in order to count until the TAR value reaches the TACCR0 value.

Configure the following registers:

```
TACCTL0 = ____________________;
TACCR0 = ____________________;
TACTL = ____________________;
```

D. Analysis of operation

After compiling the project, start the debug session and before running the application, put a breakpoint at the line of code with the `_NOP()` instruction. Go to breakpoint properties and set action to ***Write data to file***. Name the file as ***Temp.dat*** and define the data format as integer. The data starts at address 0x01040, with a length of 3C. Run the application and let the temperature data logger acquire the values for 1 hour. Use a heater or a fan to force temperature variations during the measurement period. When execution reaches the breakpoint, the file will be available in your file system. Construct a graph in Excel or a similar tool, in order to plot the temperature variation obtained by the data logger.

eZ430-RF2500 **SOLUTION**

Develop a temperature data logger through the integrated temperature sensor using the eZ430-RF2500 Development Tool.

- ❑ DCO configuration:

```
if (CALBC1_1MHZ == 0xFF || CALDCO_1MHZ == 0xFF)
  {
    while(1);           // If calibration constants erased
                        // do not load, trap CPU!!
  }

DCOCTL = CALDCO_1MHZ;  // Set DCO to 1 MHz
```

- ❑ Basic Clock module+ configuration:

```
BCSCTL1 = DIVA_3;        // ACLK = 1.5 kHz
BCSCTL3 = LFXT1S_2;      // Set VLOCLK (12 kHz)
```

- ❑ ADC10 configuration:

```
ADC10CTL1 = INCH_10 + ADC10DIV_3;    // Temp Sensor (A10),
                                     // ADC10CLK/4,
                          // ADC10 clock source: ADC10OSC,
```



```
ADC10CTL0=SREF_1 + ADC10SHT_3 + REFON + ADC10ON +ADC10IE;
            // Internal reference voltage Vref+ = 1.5 V
            // ADC10 sample-and-hold time: 64 x ADC10CLKs
            // Reference-generator voltage = 1.5 V
            // ADC10 on + ADC10 interrupt enable
```

- Timer_A configuration:

```
TACCTL0 = CCIE;         // TACCR0 interrupt enabled
TACCR0 = 1500;          // this count corresponds to 1 sec
TACTL = TASSEL_1 | MC_1 | ID_0;// ACLK, up mode to TACCR0
```

9.5.2 Lab5B: SAR ADC12 conversion

Project files

- C source file: **Chapter 9 > Lab5 > Lab5B_student.c**
- Solution file: **Chapter 9 > Lab5 > Lab5B_solution.c**

Overview

This laboratory examines the ADC12 and OA modules using the MSP-EXP430FG4618 Development Tool (MSP430FG4618 device). The test voltage is generated by the DAC12 channel 0, available in DAC12_ODAT register. The analogue signal is conditioned by the OA module (amplitude change), configured as non-inverting operational amplifier. Afterwards, this signal is applied to the ADC12 input to be converted. Compare the DAC12_ODAT and the ADC12MEM0 values.

A. Resources

The DAC12 module uses the same internal reference voltage as the ADC12 module (V_{REF+} = 2.5 V).

The OA module is configured as Non-inverting PGA with unity gain. The Non-inverting input is the DAC0 internal while the output is connected to internal/external A1 of the ADC12. The ADC12 sample-and-hold time is configured to be 64 ADC12CLK cycles. It performs a single-channel, single-conversion using ADC12OSC/1 as the clock source.

The resources used by the application (following the signal modification steps) are:

- DAC12;
- OA;
- ADC12;

- Timer_A;
- Interrupts.

B. Software application organization

The laboratory is organized following its working flow chart:

- Peripheral initialization phase, finishing with the MSP430 in LPM3;
- ISR phase, consisting of a Timer_A overflow service routine that triggers a new ADC12 conversion and it is responsible by the end of conversion.

The application starts by stopping the Watchdog Timer.

The system clock is configured by the FLL+ at 4.199304 MHz (128 x 32768Hz).

The DAC12 module is configured to present a null voltage (0 V) at the output. It uses the ADC12 internal 2.5 V reference voltage. The DAC12's output is configured with 12-bit resolution, in straight binary. DAC12 uses the full-scale output with a Medium speed/current.

The OA module is configured as non-inverting PGA, the input signal (DAC0 internal) being in the rail-to-rail range. The output of the OA is connected to internal/external A1.

The ADC12 is configured to perform a single-channel (channel A1), single-conversion. The configuration includes the activation of the same internal reference voltage as the DAC12. The ADC12 clock source is ADC12OSC, with the sample-and-hold time selected as 64 ADC12CLK cycles.

The Timer_A is configured to use the ACLK as the clock source. It will count in continuous mode (TACCR0 counts up to 0FFFFh) and generate an interrupt to update the ADC12MEM. When the interrupt is serviced, the MSP430 enters into LPM3.

C. System configuration

- **ADC12 configuration**

The ADC12 module is configured in order to have the following characteristics:

- Single-channel, single-conversion operation;
- Uses the internal signal V_{REF+} (2.5 V) as reference voltage;
- The sample-and-hold time must be 64 ADC12CLK cycles;
- The conversion result must be available on ADC12MEM0;
- The sample-and-hold clock source is defined by software.


```
ADC12CTL1 = _____________________;
ADC12CTL0 = _____________________;
```

The ADC12 module operates with reference voltages: $V_{R+} = V_{REF+}$ and $V_{R-} = AV_{SS}$. The channel selected to perform the analogue-to-digital conversion is channel A1. This channel is internally connected the OA0's output.

```
ADC12MCTL0 = ___________________;
```

❑ OA0 configuration

The OA module of the MSP430FG4168 has three operational amplifiers with wide utilization flexibility. For this laboratory it is set up using the OA0 in non-Inverting PGA mode with the following configuration:

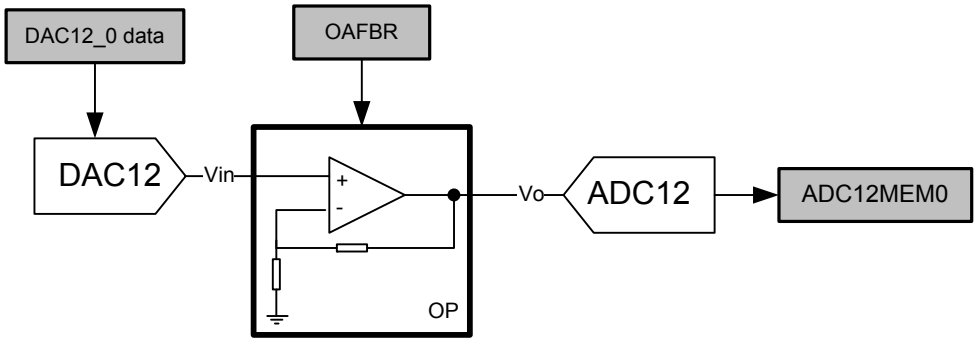
- The inverting input is connected to the DAC12 channel 0;
- The amplifier gain is configured as unity;
- The input is configured in rail-to-rail mode;
- The output is connected to the channel A1.

```
OA0CTL0 = _____________________;
OA0CTL1 = _____________________;
```

D. Analysis of operation

This laboratory uses the previous modules to construct an analogue signal chain as shown in *Figure 9-68*.

Figure 9-68. Analogue signal chain structure.



The input voltage V_{IN} is in the range 0 V and 2.5 V, with a resolution of:

$$\Delta V_{in} = \frac{2.5(V_{ref})}{2^{12}} \approx 0.6\ mV$$

The V_{IN} value is controlled by the value in the DAC12_0DATA register.

The output voltage V_o has the same characteristics as the input voltage, but scaled by a multiplication factor (gain), attributed by the OA. The OA gain is selectable through the OAFBR field in the OA0CTL1 register.

The V_o conversion result is stored in the ADC12MEM0 register.

Once the signal chain modules are configured in accordance with the previous steps, initiate the experiment by completing the file **Lab5b_student.c**, compiling it and running it on the Experimenter's board. For the evaluation of the peripherals discussed during this laboratory, set a breakpoint on the ADC12_ISR and perform the following operations:

- Configure the DAC12_0DATA register with the value 0xFF. With the aid of a voltmeter, measure the analogue input voltage A6 (DAC12 channel 0 output). The value should be in the region of 0.15 V;
- Measure the input voltage A1 (OA0's output). The voltage value should be the same;
- Execute the code. Verify the ADC12's conversion result. The value should be similar to the one of the DAC12_0DATA register;
- Double the amplifier gain (2x). Verify the voltage at A0. It should be the double of the input voltage A1 (OA0's output) given in step 2;
- Execute the code. Verify the ADC12's conversion result. The value should be two times the value of the DAC12_0DATA register;
- Execute further modifications in order to evaluate the digital-to-analogue and analogue-to-digital conversion. Do not exceed the V_o maximum value (2.5 V).

MSP-EXP430FG4618 **SOLUTION**

Using the MSP-EXP430FG4618 Development Tool, perform the signal acquisition of the DAC12's output using the integrated operational amplifiers for signal conditioning and ADC12 to convert the analogue signal to digital.

- DAC12 configuration:

```
DAC12_0DAT = 0x00;                  // DAC_0 output 0V


DAC12_0CTL = DAC12IR | DAC12AMP_5 | DAC12ENC;
//DAC_0 -> P6.6
//DAC_1 -> P6.7
```



```
//DAC reference Vref
//12 bits resolution
//Immediate load
//DAC full scale output
//Medium speed/
//Straight binary
//Not grouped
```

- ADC12 configuration:

```
ADC12CTL0 |= SHT02 | REF2_5V | REFON | ADC12ON | ENC |
ADC12SC;
//SHT1x (Sample-and-hold time) = 0000b -> N/A
//SHT0x (Sample-and-hold time) = 0010b -> 64 ADC12CLK
//MSC (Multiple sample and conversion) = 0b -> N/A
//REF2_5V (Reference generator voltage) = 1b -> 2.5 V
//REFON (Reference generator on) = 1b -> Reference on
//ADC12ON (ADC12 on) = 1b -> ADC12 on
//ADC12OVIE (overflow-int. enable) = 0b -> disabled
//ADC12TOVIE (conversion-time-overflow int enable)
// = 0b -> disabled
//ENC (Enable conversion) = 0b -> enable configuration
//ADC12SC (Start conversion)  = 1b -> Start conversion


ADC12CTL1 = CSTARTADD_0; // Start MEM0, TB1, Rpt Sing.
//CSTARTADDx (Conv. start address.) = 0000b
// -> ADC12MEM0
//SHSx (Sample-and-hold source) = 00b -> ADC12SC bit
//SHP (Sample-and-hold pulse-mode select) = 0b
// -> SAMPCON signal is sourced from the sample-input
// signal.
//ISSH (Invert signal S-H) = 0b -> not inverted.
//ADC12DIVx (ADC12 clock divider) = 000b -> /1
//ADC12SSELx (ADC12 clock source) = 00b -> ADC12OSC
//CONSEQx (Conversion sequence mode) = 00b -> Single-
// channel, single-conversion
//ADC12BUSY (ADC12 busy) = xb -> read only
```

```
ADC12MCTL0 = INCH_1 | SREF_1;
//EOS (End of sequence) = 0b -> Not Used
//SREFx (Select ref.) = 001b -> VR+=VREF+/VR-=AVSS
//INCHx (Input channel select) = 0001b  ->  A1
```

- OA configuration:

```
OA0CTL1 |= OAFC_4 | OAFBR_0;
//OAFBRx (feedback resistor) = 000b -> Tap 0 (G=1)
//OAFCx (OAx function) = 100b -> Non-inverting PGA
//OARRIP = 0b ->  OAx input range is rail-to-rail


OA0CTL0 |= OAP_2 | OAPM_3 | OAADC1;
//OANx (Inverting input) = XXb -> not important
//OAPx (Non-inverting input) = 10b -> DAC0 internal
//OAPMx (Slew rate select) = 11b -> Fast
//OAADC1 (OA output) = 1b -> output connected to A1
//OAADC0 (OA output) = 0b -> output not connected A12
```

- ADC12 ISR:

```
#pragma vector=ADC12_VECTOR
__interrupt void ADC_ISR(void)
{
   int x;
    x = ADC12MEM0;          // Reads data
    ADC12CTL0 |= ADC12SC;  // Start new conversion
}
```

- Timer_A ISR:

```
#pragma vector=TIMERA1_VECTOR
__interrupt void TimerA_ISR (void)
{
    ADC12CTL0 &= ~ADC12SC; //start new conversion
    TACTL &= ~TAIFG;
}
```


9.5.3 Lab5C: SD16_A ADC conversion

Project files

- C source file: **Chapter 9 > Lab5 > Lab5C_student.c**
- Solution file: **Chapter 9 > Lab5 > Lab5C_solution.c**

Overview

This laboratory implements a temperature data logger using the hardware kit's integrated temperature sensor. The device is configured to perform a data acquisition once every minute for one hour. Each temperature's (ºC) value is transferred to flash info memory segment B and C. When the microcontroller is not performing any task, it enters into low power mode.

A. Resources

The SD16_A module uses V_{REF+} = 1.2 V as reference voltage.

It is necessary to select the channel 6 of the SD16_A to use the integrated temperature sensor as an input. Timer_A generates an interrupt once every second, which starts conversion on the SD16_A. At the end of conversion, an interrupt is requested by the converter and the temperature value is written to flash memory.

The voltage value is converted into temperature using the mathematical expression provided in the eZ430-F2013 data sheet. After transferring the value to the flash memory, the system returns to low power mode LPM3.

The resources used by the application are:

- SD16_A;
- Timer_A;
- Ports I/O;
- Interrupts;
- Low power mode.

B. Software application organization

The application starts by stopping the Watchdog Timer.

System tests for the presence of calibration constants in info memory segment A. The CPU execution will be trapped if it does not find this information.

The digital controller oscillator (DCO) is set to 1 MHz to provide clock sources for MCLK and SMCLK, while the Basic Clock System+ is configured to set ACLK to 1.5 kHz.

The controller's flash timing is obtained from MCLK, divided by three to comply with the device specifications.

Port P1.0 is configured as output and will blink the LED once every second.

The SD16_A is configured to use the input channel corresponding to the on-chip temperature sensor (channel A6). The configuration includes the activation of the internal reference voltage: V_{REF+} = 1.2 V and the selection of SMCLK as clock signal. The converter is configured to perform a single conversion in bipolar mode and offset binary format. At the end of conversion an interrupt is requested.

The Timer_A is configured to generate an interrupt once every second. ACLK/8 is selected as the clock signal using VLOCLK as clock source and will count until it reaches the TACCR0 value (up mode). The system enters into low power mode and waits for an interrupt.

Flash memory pointers and interrupt counters are initialized. The Timer_A ISR increments variable counter and when this variable reaches the value 60 (1 minute), the software start of conversion is requested. At the end of this ISR, the system returns to low power mode.

When the SD16_A ends the conversion, an interrupt is requested. While variable ***min*** is lower than 60, the temperature is written in flash memory. The memory pointer is increased by two (word). When ***min*** = 60, the system stops operation.

C. System configuration

❑ DCO configuration

Adjust the DCO frequency to 1 MHz by software using the calibrated DCOCTL and BCSCTL1 register settings stored in information memory segment A.

```
if (CALBC1_1MHZ == _________ || CALDCO_1MHZ == _________){
    while(1);           // If calibration constants erased
                        // do not load, trap CPU!!
  }

DCOCTL = ____________________;
```

❑ Basic Clock module+ configuration

Set MCLK and SMCLK to 1 MHz. Use the internal very low power VLOCLK source clock to ACLK/8 clock signal as low frequency oscillator (12 kHz):

```
BCSCTL1 = ____________________;
BCSCTL3 = ____________________;
```


❑ SD16_A configuration

The SD16_A's input channel is the integrated temperature sensor (A6) and it uses the signal V_{REF+} (1.2 V) as reference voltage. The SD16_A clock source is SMCLK. Configure the SD16_A to perform a single conversion and enable its interrupts. What are the values to write to the configuration registers?

```
SD16CTL = ____________________;
SD16CCTL0 = ____________________;
SD16INCTL0 = ____________________;
```

❑ Timer_A configuration

Configure Timer_A register to enable an interrupt once every second. Use the ACLK clock signal as the clock source. This timer is configured in up mode in order to count until the TAR value reaches the TACCR0 value.

Configure the following registers:

```
TACCTL0 = ____________________;
TACCR0 = ____________________;
TACTL = ____________________;
```

D. Analysis of operation

❑ Measure the temperature variation over 1 hour

After compiling the project and starting the debug session, before running the application, put a breakpoint at line of code with the `_NOP()` instruction. Go to breakpoint properties and set action to ***Write data to file***. Name the file as ***Temp.dat*** and define the data format as integer. The data starts at address 0x01040 with a length of 3C. Run the application and let the temperature data logger acquire the values over 1 hour. Use a heater or a fan to force temperature variations during the measurement period. When execution reaches the breakpoint, the file will be available in your file system. Construct a graph using Excel or a similar tool, to plot the temperature variation obtained by the data logger.

eZ430-F2013 **SOLUTION**

Develop a temperature data logger through the integrated temperature sensor using the eZ430-F2013 Development Tool.

- DCO configuration:

```
if (CALBC1_1MHZ == 0xFF || CALDCO_1MHZ == 0xFF)
  {
    while(1);           // If calibration constants erased
                        // do not load, trap CPU!!
  }

DCOCTL = CALDCO_1MHZ;  // Set DCO to 1 MHz
```

- Basic Clock module+ configuration:

```
BCSCTL1 = DIVA_3;        // ACLK = 1.5 kHz
BCSCTL3 = LFXT1S_2;      // Set VLOCLK (12 kHz)
```

- SD16_A configuration:

```
SD16CTL = SD16REFON + SD16SSEL_1;  // 1.2V ref, SMCLK
SD16INCTL0 = SD16INCH_6;           // Temp. sensor: A6+/-
SD16CCTL0 = SD16SNGL + SD16IE;  // Single conv, interrupt
```

- Timer_A configuration:

```
TACCTL0 = CCIE;         // CCR0 interrupt enabled
TACCR0 = 1500;          // this count corresponds to 1 sec
TACTL = TASSEL_1 | MC_1 | ID_0;  // ACLK, up mode to CCR0
```

9.5.4 Lab5D: Voltage signal comparison with Comparator_A

Project files

- C source files: **Chapter 9 > Lab5 > Lab5D_student.c**
- Solution file: **Chapter 9 > Lab5 > Lab5D_solution.c**

Overview

This laboratory analyses Comparator_A operation. A voltage is applied to one of the Comparator's inputs, generated either by the DAC12 or by other external source. Whenever the external voltage value is higher than the comparison value internally generated, an interrupt is generated that switches the LED state.

A. Resources

The resources used by the application are:

- DAC12;
- Comparator_A;
- Digital IO;
- Timer_A.

B. Software application organization

The application starts by stopping the Watchdog Timer.

Timer_A is configured to generate an interrupt once every msec, and updates the DAC12 output in order to provide a ramp signal.

The Comparator_A's output is configured to be accessible at pin P6.6, which is available on Header 4 pin 7. The signal applied to CA0 input is compared with 0.5 Vcc internal reference voltage. Every time that a compare match occurs, an interrupt is requested and switches the state of LED1.

C. System configuration

- **Comparator_A configuration**

Configure the registers in order to receive the external signal at the CA0 input and compare it with the internal reference 0.5 Vcc. Enable the comparator with an interrupt triggered on a low-to-high transition of the comparator output.

```
CACTL1 = _____________________;
CACTL2 = _____________________;
```

D. Analysis of operation

The experimental verification of this laboratory can be accomplished by connecting the DAC12's output, available on Header 8 pin 7, to the Comparator_A's input CA0, available on Header 4 pin 7.

Observe the signals wave form at the Comparator_A's input and output using an oscilloscope. The LED1 switches state whenever the input's voltage value is lower than the compare value.

MSP-EXP430FG4618 **SOLUTION**

Using the MSP-EXP430FG4618 Development Tool, perform a voltage signal comparison with the internal reference.

- Comparator_A configuration:

```
CACTL1 = CAON + CAREF_2 + CARSEL;        // Enable comp,
                                         // ref = 0.5*Vcc
CACTL2 = P2CA0;                          // Pin to CA0
```

9.6 Quiz

1. An ideal operational amplifier is an amplifier with:

(a) Zero Z_{IN}, infinite gain, zero Z_O, infinite bandwidth and zero offset;

(b) Infinite Z_{IN}, infinite gain, zero Z_O, infinite bandwidth and zero offset;

(c) Infinite Z_{IN}, zero gain, zero Z_O, infinite bandwidth and zero offset;

(d) Infinite Z_{IN}, infinite gain, infinite Z_O, zero bandwidth, and zero offset.

2. When R_f = 0 and R_1 = infinity, an Op-Amp becomes:

(a) An amplifier with gain equal to infinity;

(b) An amplifier whose output voltage equals its input voltage (voltage follower);

(c) All of above;

(d) None of above.

3. When Op-Amp control register bits OAFCx = 4, its topology is configured for:

(a) Unity gain buffer;

(b) Comparing OpAmp;

(c) Non-inverting PGA;

(d) Differential OpAmp.

4. To set a gain of A_{VD} = 8, the OAx feedback resistor Op-Amp control register bits, OAFBRx, must be configured as:

(a) OAFBRx = 6;

(b) OAFBRx = 3;

(c) OAFBRx = 4;

(d) OAFBRx = 7.

5. The internal connection of the OAx output to the A0 ADC12 input channel requires setting the OA control bit:

(a) OARRIP;

(b) OAADC0;

(c) OAADC1;

(d) None of above.

6. The performance of an ADC is expressed by the specifications:

(a) Speed, Accuracy and Signal-to-noise and distortion ratio (SINAD);

(b) Offset and gain errors, and Signal-to-noise ratio (SNR);

(c) Integral (INL) and Differential Nonlinearities (DNL) and Total harmonic distortion (THD);

(d) All of above.

7. Oversampling means:

(a) An ADC performance parameter of noise limit;

(b) Sampling at a rate much higher than the signal of interest;

(c) To increase the resolution;

(d) All of above.

8. A low cost, low power consuming application that requires a 12-bit resolution with a 100 Hz output data rate should use an ADC with the following architecture:

(a) Slope;

(b) Sigma-Delta;

(c) SAR;

(d) Flash.

9. The SAR gets its name from a process that:

(a) Successively compares the input analogue voltage to the output of a DAC that has a binary weighted input code for a capacitive SAR ADC;

(b) Sums a series of binary-weighted currents;

(c) Sums current from a ladder resistor network for a resistive SAR ADC;

(d) Sums voltages from a resistor string.

10. A 12-bit SAR ADC is included in the:

(a) eZ430-F2013;

(b) eZ430-RF2500;

(c) Experimenter's board;

(d) All of above.

11. To configure ADC10, present in eZ430-RF2500, to use the internal voltage generator, the control register bits:

(a) REFOUT must be set;

(b) REF2_5V must be reset;

(c) REFON must be set;

(d) REFBURST must be set.

12. For the MSP430FG4618 present in the Experimenter's board, when the SHP control register bit is set, the sample timing method of the ADC12 is configured in:

(a) Extended sample mode;

(b) Synchronization with ADC12CLK mode;

(c) Pulse mode;

(d) Fixed sample period length mode.

13. The ADC12 interrupt sources available are:

(a) ADC12IFGx and ADC12OV;

(b) ADC12TOV;

(c) All of above;

(d) None of above.

14. The configuration of the ADC12 input channel as an internal temperature sensor requires the INCHx control bits defined as:

(a) INCHx = 12;

(b) INCHx = 9;

(c) INCHx = 10;

(d) None of above.

15. A Sigma-Delta ADC presents the advantages of:

(a) High resolution and accuracy, low sample rate, low cost;

(b) High resolution and stability, low power, low cost;

(c) High resolution and stability, low power, moderate cost;

(d) High bandwidth and accuracy, high sample rate, moderate cost.

16. The quantization noise of a Sigma-Delta ADC:

(a) Limits the dynamic range;

(b) Is actually the "round-off" error;

(c) Is pushed into higher frequencies due to oversampling;

(d) All of above.

17. The SD16_A digital output configured as unipolar provides for an input voltage:

(a) $[-V_{FSR} \quad V_{FSR}] = [0000h \quad FFFFh]$;

(b) $[-V_{FSR} \quad V_{FSR}] = [8000h \quad 7FFFh]$;

(c) $[-V_{FSR} \quad V_{FSR}] = [8000h \quad FFFFh]$;

(d) None of above.

18. The SD16_A control register bits that adjusts the frequency of the notch digital filter are:

(a) SD16SSELx and SD16DIVx;

(b) SD16OSRx and SD16XOSRx;

(c) All of above;

(d) None of above.

19. Using the Comparator_A to measure resistive elements, its accuracy is limited by:

(a) Resistor precision, number of counts, capacitor precharge;

(b) Resistor precision, V_{CC} stability, capacitor precharge;

(c) RC tolerance, V_{CC} stability, V_{REF} stability;

(d) None of above.

20. To use the Comparator_A with internal low pass filtering, the control register bit:

(a) CAON must be reset;

(b) CAF must be set;

(c) CARSEL must be reset;

(d) CAEX must be set.

21. To use the Comparator_A with an internal diode reference voltage in the positive terminal, the control register bits:

(a) CAREFx = 2; CARSEL = 0; CAEX = 1;

(b) CAREFx = 1; CARSEL = 1; CAEX = 1;

(c) CAREFx = 3; CARSEL = 1; CAEX = 1;

(d) CAREFx = 0; CARSEL = 0; CAEX = 0.

Solution: 1. (b); 2. (b); 3. (c); 4. (a); 5. (b); 6. (d); 7. (d); 8. (a); 9. (a); 10. (c); 11. (d); 12. (c); 13. (d); 14. (c); 15. (b); 16. (d); 17. (a); 18. (c); 19. (a); 20. (b); 21. (c)

9.7 FAQs

1. Which are the important electrical characteristics of an ADC datasheet that relate to its performance?

(a) ADC's performance over temperature at specified speed (clock and sample frequencies)

(b) Ensured minimum and maximum values based on tests in production

(c) Clear test conditions for specified parameters

(d) No hidden and unclear limitations.

2. What is the maximum resolution of the slope ADC?

The Slope ADC resolution depends on the ratio of the Timer clock frequency and the time constant of the external RC used. The higher the counts of the Timer within the RC time constant, the better the resolution. Smaller RC time constants allow faster conversions. With proper combination of RC time constant, Timer clock frequency and integration of conversions over time, 16-bits of resolution can be achieved.

3. Why a sequence of channels analogue-to-digital conversions with ADC12 does not stop when the enable conversion control bit, ENC, is set?

In a sequence of channels conversion, it is necessary to set the EOS control bit (End Of Sequence). A procedure to stop the sequence relies on shifting to the single-channel mode and then reset ENC.

4. There is any precaution to bear in mind with the input terminal connections of the comparator during a design phase?

Even during a design phase, if the comparator is on and its input terminals are not connected, there is a chance that the floating levels at the terminals may cause unexpected interrupts and increased current consumption. It is recommended to connect the input terminal to a signal, power, or ground before enabling the comparator.